




# *TEORÍA*

# ELECTRÓNICA

# INDUSTRIAL

## ENVÍO 10



# TECNICAS DIGITALES

Paralelamente a la aparición de las computadoras nació una técnica llamada **"SISTEMAS DIGITALES"**, que sirvió de fundamentos a éstas. El avance de estos sistemas se ha desarrollado tan ampliamente que en la actualidad son aplicadas no tan solo en dispositivos, digitales, sino también en dispositivos analógicos para funciones específicas, por ejemplo, en televisores, sistemas de audio, video games, lavadoras, cocinas, hornos, computadoras, calculadoras, relojes, control retomo, etc., en fin pocos son los artefactos donde aún no se aplica esta técnica. El término analógico deriva del griego **"análogo",** que significa: Algo que escomo alguna otra cosa.

El término digital deriva del latín **"digitus",** que significa: Dedo.

Los sistemas analógicos son proporcionales y continuos tan uniformemente como el ruido lo permita; en cambio, los sistemas digitales desarrollan procesos que son inherentemente discontinuos. Estos procesos evolucionan por dos estados lógicos definidos como (0 – 1, alto – bajo, abierto o cerrado, etc), que pueden ser representado por pulsos de voltaje, corriente, magnetismo, etc.

Cuando estos pulsos son debidamente contabilizados se pueden realizar operaciones tales como, suma, resta, multiplicación, división, almacenamiento de datos etc.

Los sistemas lógicos de dos estados se llaman sistemas binarios y el término binario significa: compuesto de dos partes diferentes.

Cuando se elige uno de los dos estados lógicos, se tiene un digito binario al cual se le denomina bit **(BINARY DIGIT)**, que es la unidad de información más pequeña dentro de los sistemas digitales.

- **1 BYTE = 8 BIT**
- **1 KILO BYTE (Kb) = 1024 BYTE (128 x 8 = 1024)**
- **1 MEGA BYTE (Mb) = 1048576 BYTE (1024 X 1024)**
- **1 GYGA BYTE (Gb) = 1048576 x 1024 BYTE**

De este breve análisis se deduce que los sistemas digitales tienen un rol de primera importancia en la electrónica moderna, y por motivo resulta impredecible comprender el sistema binario y conocer su aritmética de los dispositivos de tipo lógico, como además el conocimiento de otros sistemas numéricos de base diferente que nos permitan comprender el lenguajes de los sistemas digitales de máquinas microprocesadoras, y microcomputadores.

## FORMA DE REPRESENTAR UN NUMER O

La base o raíz de un sistema numérico define la cantidad de dígitos que usa dicho sistema, por ejemplo, el sistema decimal opera con 10 dígitos entre (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).

En cualquier sistema podemos asignar un peso a cada digito y así repr esentar un sistema pesado. Ejemplo 345 (el valor 10 como subíndice indica en este caso la base del sistema que es 10).

Este número puede ser representado también del siguiente modo:

**3 x 100 + 4 x 10 + 5 x 1 = 345**

o también como potencia de 10.

**3 x 10 + 4 x 10 + 5 x 10 = 345**

Observará que cada digito se le asigna un peso que corresponde a 10 para las unidades, ya que 10 = 1; para las decenas ya que 10 = 10; 10 para las centenas ya que 10 = 100, etc..

Otro sistema numérico es el binario el cual tiene b ase 2, ya que dispone de dos dígitos (0 y 1) para representar las cantidades.

## CONVERSIÓN DECIMAL A BINARIA

Para convertir un valor decimal a binario se divide el decimal a sucesivamente por 2 para obtener cuocientes enteros solamente, división que se pr olonga hasta obtener un resto 0 ó 1. El primer resto encontrado lo llamaremos el bits menos significativo (bms) y el último resto que siempre será 1 lo llamaremos el bits más significativo (bms).

**Ejemplo**

**Ejercicios:** Hallar el equivalente binario de los siguientes números decimales.

133=________________ 68=_____________ 126=_____________ 85=___________

400=________________ 18=_____________ 106=_____________ 34=___________

## CONVERSIÓN DE BINARIO A DECIMAL

Cada dígito o bits tiene asignado un peso en base 2 de orden creciente a partir del bit menos significativo. Para efectuar la transformación a decimal sumaremos todos los pesos para los que el bits tome el valor 1.

**Ejemplo**

| | |
|---|---|
| 16 8 4 2 1 | 16 |
| | 4 |
| 2 2 2 2 2 | 2 |
| | $22_{10}$ |
| 1 0 1 1 0 | |

$10110_2 = 22_{10}$

**Ejercicios:** Transformar a decimal los siguientes números binarios

$10101_2 =$

$1111_2 =$

$11100_2 =$

$1001_2 =$

Binarios con signos: Normalmente en el lenguaje de microprocesadores se oper a con datos de 8 bits, lo que permite representar números sin signos o resta hasta 255. Pero cabe la necesidad de representar números negativos o positivos convencionalmente en un sistema numérico de 8 bits, el octavo bits se usa para la representación de l signo siendo esta positivo cuando el bits 7 es 0, y negativo cuando el bits 7 es 1.

**Ejemplo**

7 6 5 4 3 2 1 0

0 0 0 0 0 0 0 1 Representa $+1_{10}$

1 0 0 0 0 0 0 1 Representa $-1_{10}$

0 1 1 1 1 1 1 1 Representa $+127_{10}$

1 1 1 1 1 1 1 1 Representa $-127_{10}$

**Bits Signo**

Como se observa usando representaciones con signos, el número mayor que se pueda representar es 127 , por lo que representa cantidades mayores será necesario aumentar la cantidad de bits a utilizar en la representación. Por ejemplo, usando 16 bits se puede representar números con signos hasta $32768_{10}$.

## SISTEMAS NUMERICOS UTILIZADOS EN ELECTRONICA DIGITAL

| DECIMAL | BINARIOS NATURAL | OCTAL | HEXADECIMAL |
|---|---|---|---|
| 0 | 0000 | 0 | 0 |
| 1 | 0001 | 1 | 1 |
| 2 | 0010 | 2 | 2 |
| 3 | 0011 | 3 | 3 |
| 4 | 0100 | 4 | 4 |
| 5 | 0101 | 5 | 5 |
| 6 | 0110 | 6 | 6 |
| 7 | 0111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |
| 16 | 10000 | 20 | 10 |
| 17 | 10001 | 21 | 11 |
| 18 | 10010 | 22 | 12 |
| 19 | 10011 | 23 | 13 |
| 20 | 10100 | 24 | 14 |
| 21 | 10101 | 25 | 15 |
| 22 | 10110 | 26 | 16 |
| 23 | 10111 | 27 | 17 |
| 24 | 11000 | 30 | 18 |
| 25 | 11001 | 31 | 19 |
| 26 | 11010 | 32 | 1 A |
| 27 | 11011 | 33 | 1 B |
| 28 | 11100 | 34 | 1 C |
| 29 | 11101 | 35 | 1 D |
| 30 | 11110 | 36 | 1 E |
| 31 | 11111 | 37 | 1 F |

## NUMEROS HEXADECIMALES

El sistema hexadecimal de números es el sistema de números de base 16, que utiliza los símbolos 0-9, A, B, C, D, E, F.

La ventaja del sistema hexadecimal es su facilidad de conversión directa a un número binario, de cuatro bit en la tabla numérica se puede observar que cada número binario de cuatro bit, o sea, se puede observar que cada número binario de 4 bit, o sea, del 0000 al 1111, puede representarse por un solo digito hexadecimal.

Al fijarse en la columna decimal de la tabla numérica se puede ver que el equivalente de 16 en el sistema hexadecimal es 10 lo que demuestra que el sistema hexadecimal también emplea el concepto de valor posición. El 1 en $10_{16}$
+ significa 16 unidades, mientras que el 0 representa cero unidades.

## CONVERSIÓN DE HEXADECIMAL A DECIMAL

Para tal efecto, se confecciona un sistema pesado, en el que cada digito o bits tiene asignado un peso en base 16 da orden creciente a partir del bits menos significativo el que se multiplica por cada símbolo del sistema hexadecimal, luego los valores parciales que se obtengan se suman para, obtener la conversión a decimal.

**Ejemplo**

**Ejercicios:** Convertir a decimal los siguientes números hexadecimales.

$43C_{16}$ = $FF_{16}$ = $248_{16}$ =

## CONVERSIÓN DE DECIMAL A HEXADECIMAL

Se divide el número decimal por 16 en fora sucesiva hasta obtener un resto final menor que 16.

**Ejemplo**

```
          584    :  16 = 36
              10

          104
bms         08

          36     :  16 = 2
            4

            2    :  16 = 0
bms         2
```

**Ejercicios:**

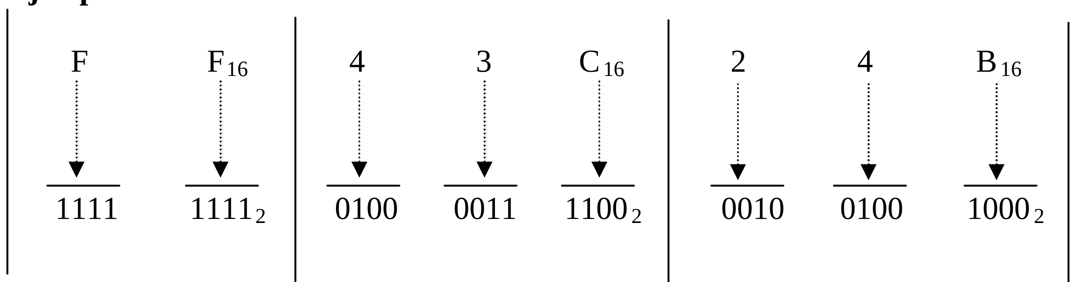
$1084_{10} =$

$255_{10} =$

## CONVERSIÓN DE HEXADECIMAL A BINARIO

En este caso para la conversión diremos que cada digito hexadecimal será representado en su equivalente binario, pero empleando en cada caso 4 bits.

**Ejemplo**




## CONVERSIÓN DE BINARIO A HEXADECIMAL

Para la conversión debemos separar el número binario en grupos de 4 bits empleando desde el bits menos significativo, y luego representando cada grupo por su equivalente en hexadecimal.

**Ejemplo**

| | | |
|---|---|---|
| 1100 | 1010 | $1001_2$ |
| C | A | 916 |

$110010101001_2 = CA9_{16}$

| | | |
|---|---|---|
| 11 | 0110 | $1110_2$ |
| 3 | 6 | E16 |

$110111011102 = 36E_{16}$

## ARITMÉTICA BINARIA

El sistema binario de numeración admite las mismas operaciones aritméticas que el sistema decimal, es decir que podrán realizarse la suma, resta, multiplicación y división de números binarios.

Para nuestro propósito tienen sumo interés las operaciones aritmáticas.

## 1.- SUMA DE NUMEROS BINARIOS

Para efectuar la suma binaria basta tener presente las sigui entes reglas:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 con reserva de 1
1 + 1 + 1 = 1 con reserva de 1

**Ejemplo:**

```
      1 1  (reserva)                    1 1 1 1  (reserva)
        1  1  0  1      13(10)            1 1 1 1     15(10)
        1  1  1  0      14(10)              1 1 1      7(10)
 +  ____________      ______        +  _________    ______
      1  1  0  1  1(2)  27(10)          1 0 1 1 0(2)  22(10)
```

**Ejercicios:** Efectuar las siguientes sumas.

```
  1  0  0  1-----   9                1  0  0  1  1-----   19
+    1  0  1-----+  5              + 1  1  0  0  1-----+  25

  1 0  1  1  1-----23
+1  1  0 1  1-----27
```

Cuando la cantidad de sumandos es mayor que dos, se procede a seccionar la suma en sumas parciales de dos sumandos y luego hacer lo mismo con los resultados parciale s, hasta llegar al resultado final.

```
  1  1  0  1  1        (1)
  1  0  1  1  0        (2)
  1  0  1  1  1        (3)
+ 1  1  1  1  0        (4)
```

**2ª UNIDAD: SISTEMAS DE LÓGICA COMBINACIONAL**

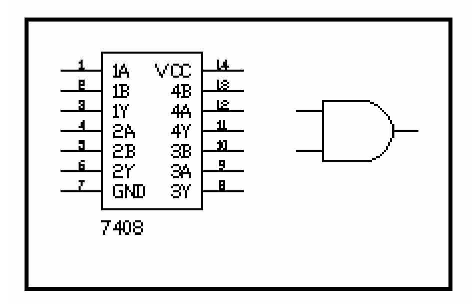
## COMPUESTAS LOGICAS BASICAS Y TABLAS DE VERDAD

Todo sistema digital, sea combinacional o secuencial, se encuentra implementado a partir de tres circuitos u operadores básicos, denominados de acuerdo a su función AND, OR y NOT. Esto significa que si comenzamos a descomponer un sistema en partes cada vez más elementales, tendremos finalmente uno cuya arquitectura corresponderá a un conexionado de operadores lógicos básicos.

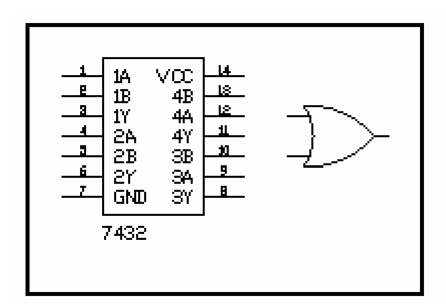
Estos operadores básicos se definen como sistemas electrónicos que basándose en la numeración binaria permiten el desarrollo de múltiples operaciones matemáticas, almacenamiento de datos, control exacto del tiempo, etc.

Las compuestas se pueden clasificar en:

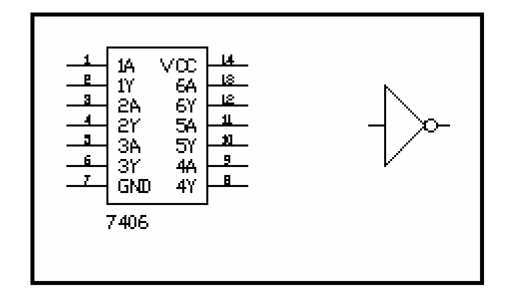
**a)Compuerta u operación AND :**




**b) Compuerta u operador OR :**



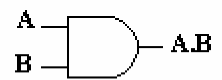

**c) Compuerta u operador NOT :**



Cada compuerta está diseñada para una función determinada y esa función se puede comparar mediante una tabla que indica todas las posibilidades combinaciones de entrada y su salida. Esta tabla recibe el nombre de "Tabla de verdad".

La cantidad de combinaciones que es capaz de manejar una determinad a compuerta lógica dependerá del número de entradas ; por ejemplo, si la compuerta en cuestión consta de dos entradas, la base del sistema que es 2 se eleva a un exponente igual a su número de entradas:

**<u>*Para una compuerta de dos entradas:</u>**

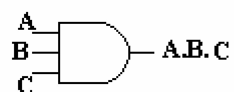



$2^2$= 2.2 = 4 combinaciones de entrada

**Combinaciones de entrada**

| A | B | A.B |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

**<u>*Para una compuerta de tres entradas:</u>**




$2^3$= 2.2.2 = 8 combinaciones de entrada

**Combinaciones de entrada**

| A | B | C | A.B.C |
|---|---|---|---|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

Para comprender el funcionamiento de las compuertas lógicas, también se hace necesario entender que:

a) Un pulso negativo es igual a “0”
b) Un pulso más negativo que otro es “0”
c) Un pulso positivo es igual a “1”
d) Un pulso más positivo que otro es igual a “1”

**Lógica Positiva**

a) Un pulso negativo es igual a “1”
b) Un pulso más negativo que otro es igual a “ 1”
c) Un pulso positivo es igual a “0”
d) Un pulso más positivo que otro es igual a “0”

**Lógica Negativa**

## DEFINICIONES FUNDAMENTALES

**VARIABLE BINARIA**: Es aquella que puede tomar solo dos valores distintos. Estos valores pueden designarse de diferentes f ormas:

**a) “1” ó “0”**
**b) “alto” ó “bajo”**
**c) “si”ó “no”**
**d) “cierto” ó “falso”**

**Ejemplo de estados binarios:**

a) El estado de un interruptor puede ser abierto o cerrado.
b) El estado de una lámpara puede ser encendida o apagada.
c) El estado de un transistor puede ser saturado o al corte.

## FUNCION LOGICA.-

Una función lógica puede ser dependiente de una ó más variables, pero no puede aceptar más que dos valores: “1” ó “0”.

La función lógica no queda definida si no se precisa que valor adopta para cada combinación posible de los valores de las variables.

## TABLA DE VERDAD

Otra forma de expresar una función es mediante una tabla de verdad. La tabla de verdad es una forma comoda de resumir los valores de una función para todas las combinaciones posibles de las variables de entrada. Si la función depende de n cantidad de variables de entrada, habra combinaciones posibles para los valores de las variables. La función se

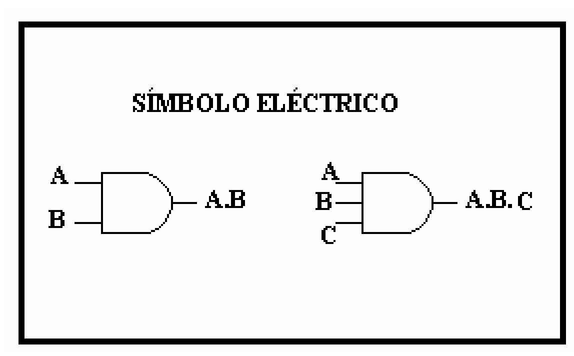
define por una tabla de verdad de varias columnas (una por cada variable y una para la función) y varias líneas (una para cada combinación de valores).

| E1 | E2 | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## COMPUERTA Y (AND)

La característica principal de esta compuerta es que para tener una salida "1", todas sus variables de entrada deben valer "1".

Esta compuerta puede tener dos ó más entradas, pero una salida.



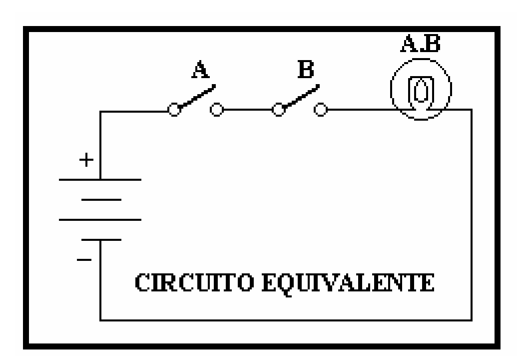




**Tabla de Verdad**

| **ENTRADAS** | | **SALIDA** |
|---|---|---|
| **A** | **B** | **Y = A.B** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| | | |

Esta compuerta es equivalente a dos ó más interruptores conectados en serie, de modo que para que circule la señal o corriente es necesario que todos los interruptores estén cerrados, pues basta que este abierto uno solo para que la corriente se interrumpa.

Si designamos por uno al estado encendido y por cero el de apagado, y por uno al estado de conexión y por cero al estado de desconexión de los interruptores, tendremos la tabla de verdad mostrada en la figura.

En la práctica el cero binario se define como un voltaje bajo o masa y el uno binario se define como un voltaje alto, el cual puede ser de 3 – 5 ó más volts.

El álgebra booleana es una forma simbolica que muestra como operan los circuitos lógicos, mediante un método taquigrafico.

La expresión booleana para la compuerta AND indicada anteriormente es:

**F = A . B** **ó** **F = AB**

Esto se lee, A y B igual F . El punto entre las dos letras no indica multiplicación.
Las leyes formales del álgebra booleana para la compuerta AND son:

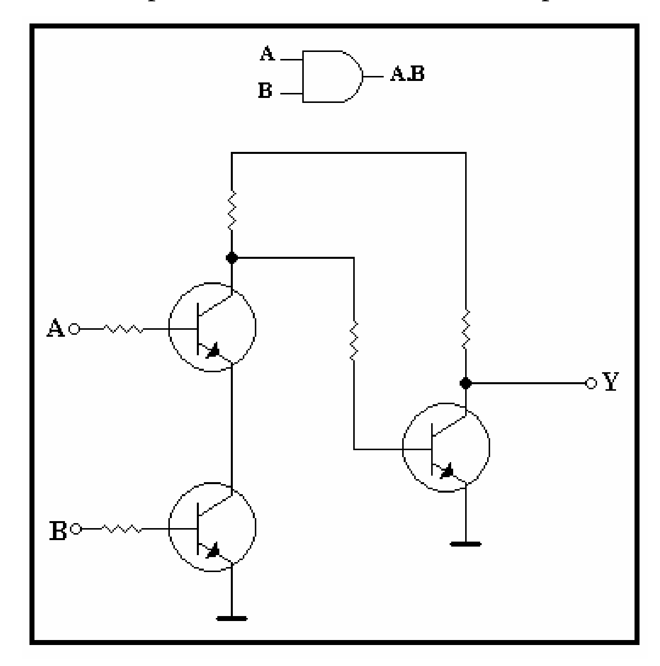
$$A \cdot \overline{A} = 0$$
$$A \cdot 1 = A$$
$$A \cdot 0 = 0$$
$$A \cdot A = A$$

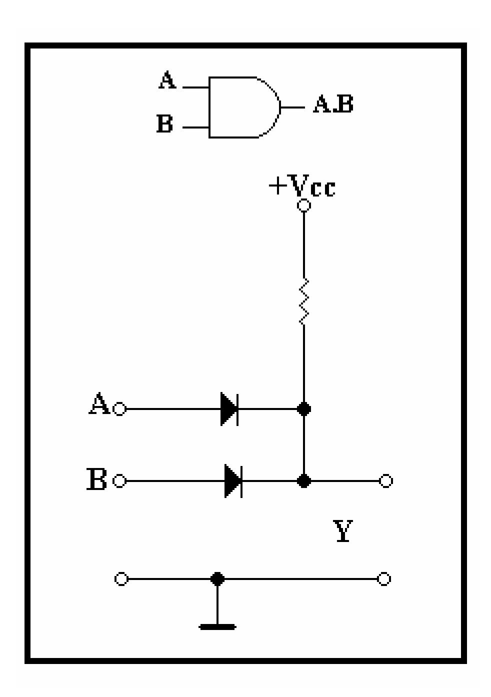
La barra sobre la variable A representa el complemento de A(opuesto) y se lee “A negado”. En álgebra de BOOLE el complemento de cero es uno y el de uno es cero. Por lo tanto:

$$A = \overline{A}$$
$$\overline{1} = 0$$
$$\overline{0} = 1$$

En la figura se muestra la tabla de la verdad de la función AND.

El operador que realiza la función AND se puede implementar eléctricamente de muchas formas diferentes, tal como lo muestra la figura siguiente, donde los estados eléctricos considerados en la operación de los circuitos corresponden a niveles de voltaje.

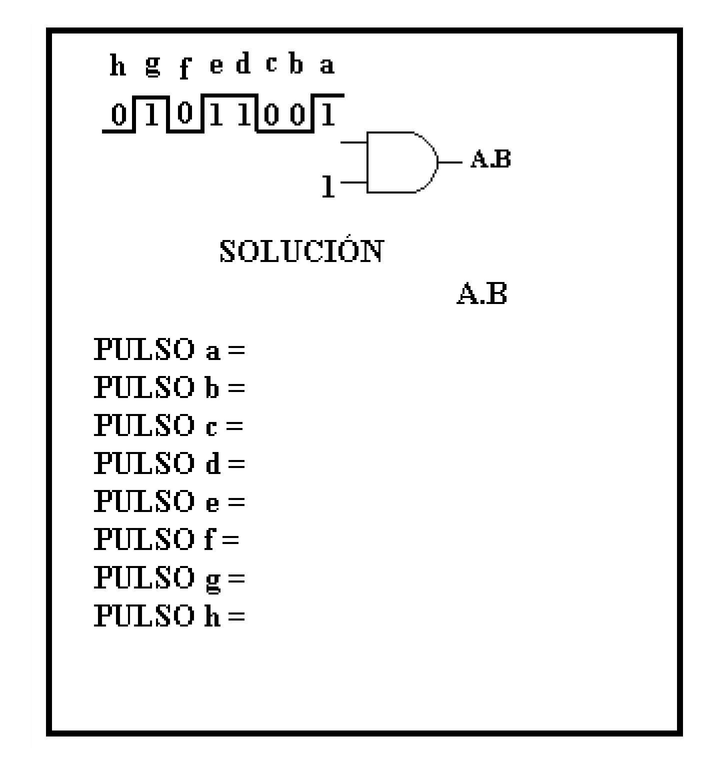


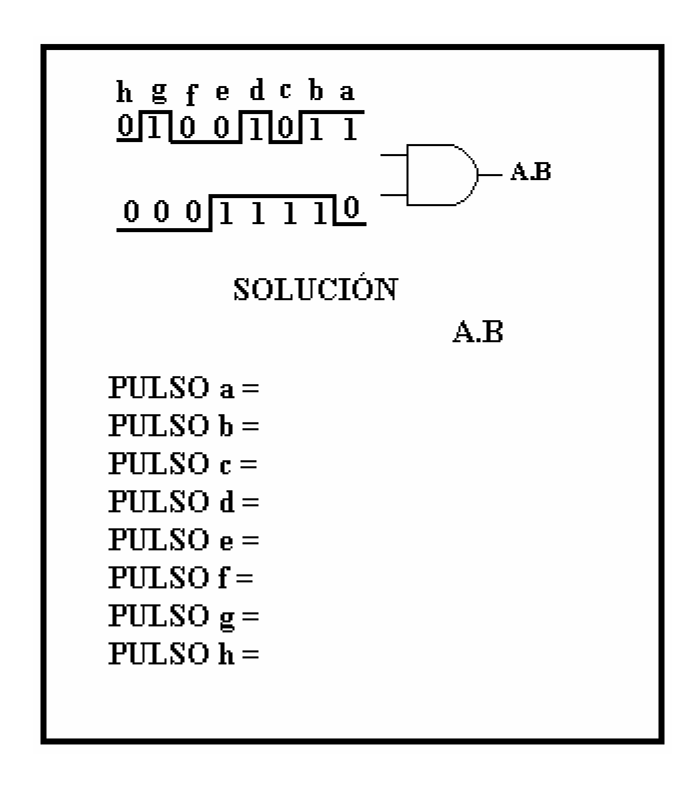



| A | B | F |
|---|---|---|
| 0V | 0V | 0V |
| 0V | Vcc | 0V |
| Vcc | 0V | 0V |
| Vcc | Vcc | Vcc |

**Ejercicios:**

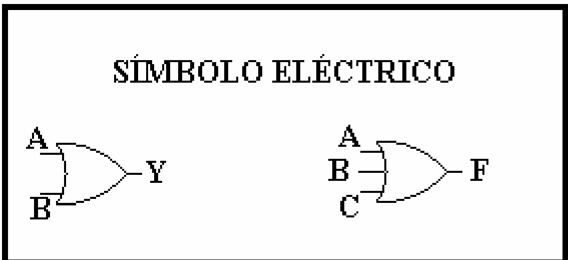
Indique cuales serán los pulsos de salida en las siguientes compuertas AND:



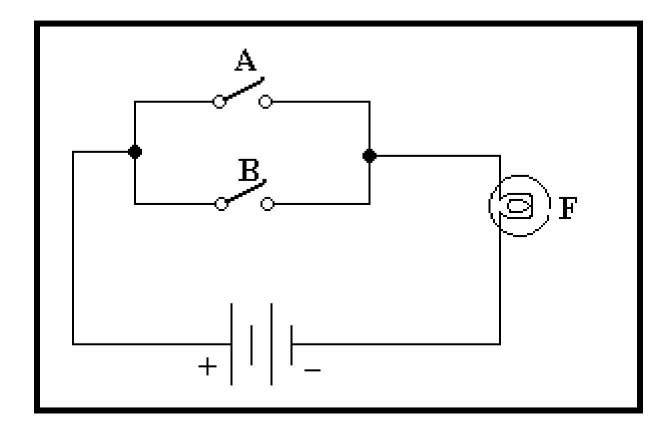



## COMPUERTA O (OR)

Esta compuerta se caracteriza por tener salida 1 cuando una ó todas sus entradas valen 1.




Esta compuerta es equivalente a dos interruptores conectad os en paralelo, de modo que para que circule la corriente basta que uno cualquiera de ellos se encuentre cerrado.



La expresión booleana para la compuerta 0 (OR) indicada anteriormente es:

**Y = A + B**

Esto se lee como A o B es igual a Y. E1 signo más en este caso no indica suma.

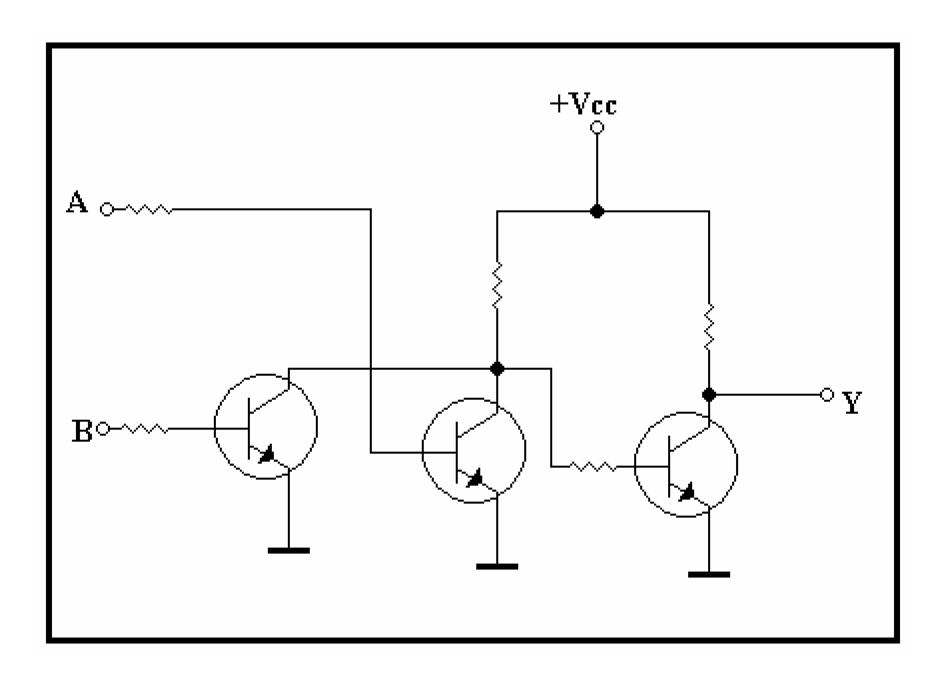
La compuerta **0 (OR**), al igual que la **Y (AND)**, puede tener varias entradas pero una salida.

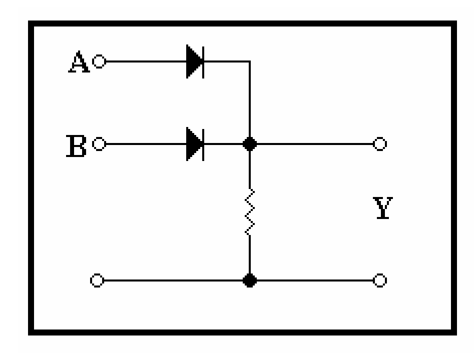
En la figura anterior se muestra un circuito formado por dos interruptores conectados en paralelo y en serie con una lámpara indicadora. De acuerdo con la configuración del circuito, la lámpara encenderá si se cierra el interruptor A ó el B o ambos interruptores. La tabla de verdad muestra la situación del circuito, para cada una de las posibles combinaciones de entrada.

**Tabla de la verdad**

| **ENTRADAS** | | **SALIDA** |
|---|---|---|
| **B** | **A** | **B+A** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Al igual que la compuerta **Y (AND)**, la función **0 (OR)** se puede obtener por distintos medios electrónicos, tal como se muestra en la figura siguiente:

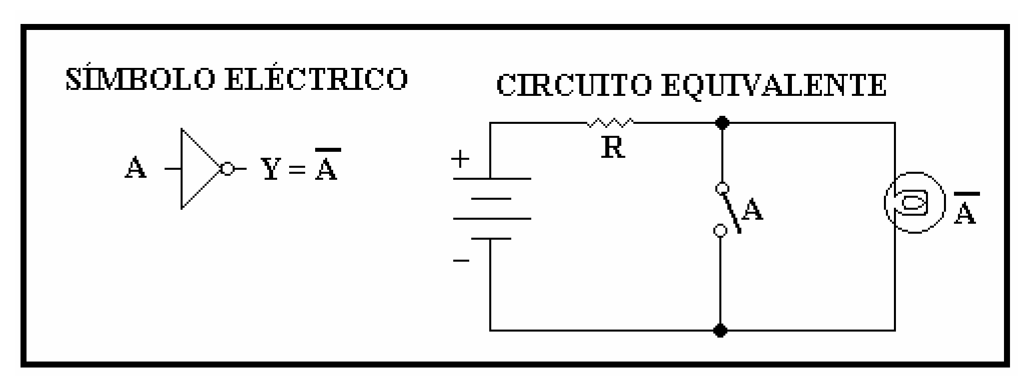





| A | B | Y |
|---|---|---|
| 0V | 0V | 0V |
| 0V | Vcc | Vcc |
| Vcc | 0V | Vcc |
| Vcc | Vcc | Vcc |

## COMPUERTA NO (NOT)

A esta compuerta también se le conoce con el nombre de “INVERSOR” ó “NEGADOR” y se caracteriza por tener siempre en su salida un estado lógico opuesto al de su entrada.



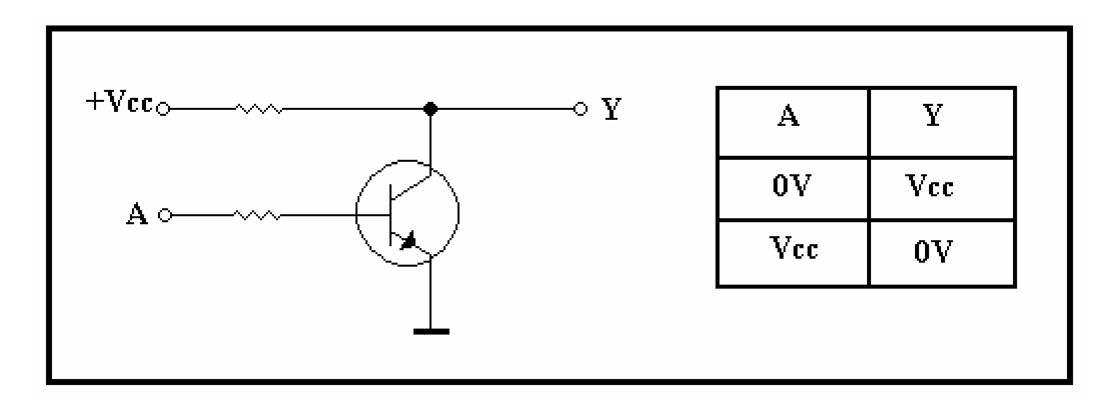

La configuración circuital de este operador lógic o se muestra en la figura siguiente, donde tenemos un circuito formado por un transistor en montaje emisor común. Supongamos que la base del transistor no posea polarización, el transistor se mantendrá en la condición de corte, por lo cual el voltaje de s alida se mantendrá en un valor cercano al de Vcc; por lo contrario, cuando la base del transistor posea polarización, el voltaje de salida, que corresponde al existente entre colector y emisor se mantendrá en un valor cercano a cero volts.

**Ecuación de Salida**
**F = A**

**Tabla de la verdad**

| ENTRADA | SALIDA |
|---|---|
| **A** | **Y** |
| 0 | 1 |
| 1 | 0 |

Resumiendo, podemos apreciar que a la situación de la salida de este circuito correspondiente siempre a la condición opuesta a la existente en la entrada. Esto queda de manifiesto en la tabla de verdad que se muestra junto a la configuración circuital.



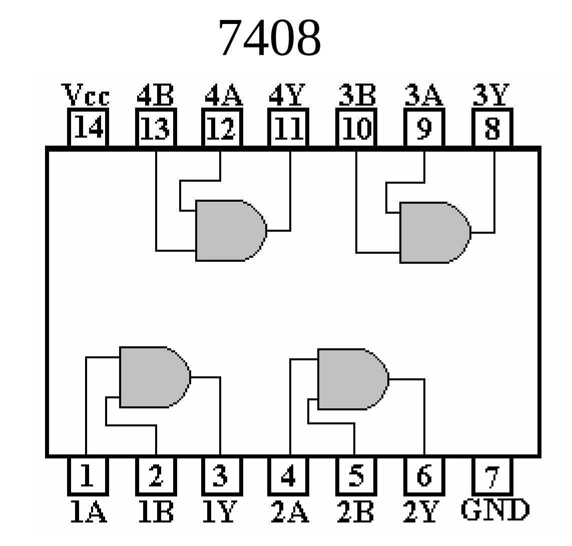

| A | Y |
|---|---|
| 0V | Vcc |
| Vcc | 0V |

Como ya dijimos en un principio, todo sistema digital, cualquiera sea su envergadura, se encuentra implementado a partir de estos tres operadores lógicos básicos, los cuales al ser conectados en distintas configuraciones, darán forma a nuevos sistemas capaces de realizar otras funciones lógicas.

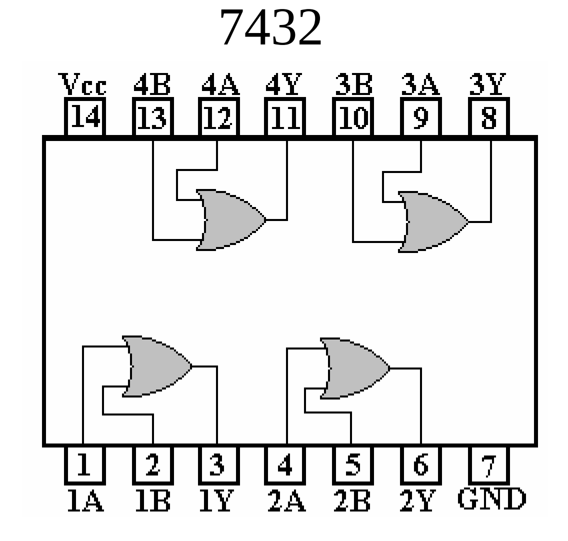
Cuando se implementan sistemas digitales, no es necesario considerar los aspectos electrónicos de la operación del sistema, esto debido a que la tecnología actual de alta integración permite que en un solo circuito integrado (CHIP) puedan “colocarse” todos los componentes electrónicos que constituyen a un sistema digital que realice una función específica; así, por ejemplo: existen circuitos integrados que contienen los operadores lógicos básicos que hemos estudiado.

En la figura siguiente se muestran los circuitos integrados TTL 7408, 7432 y 7404, que contienen los operadores AND, OR y NOT respectivamente.

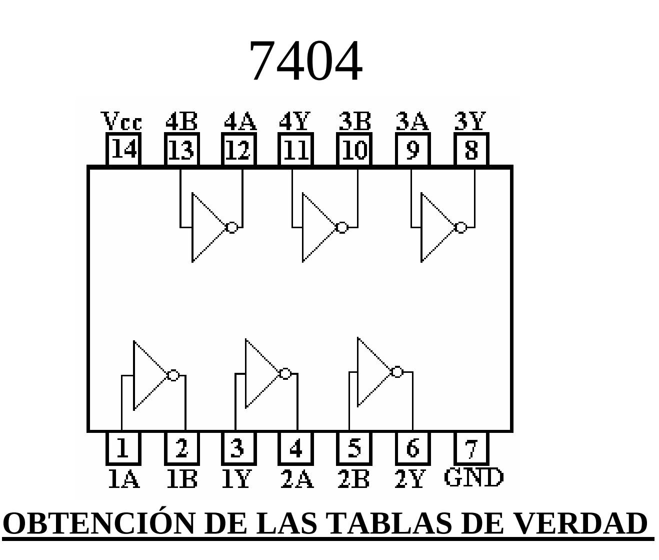
7408




7432



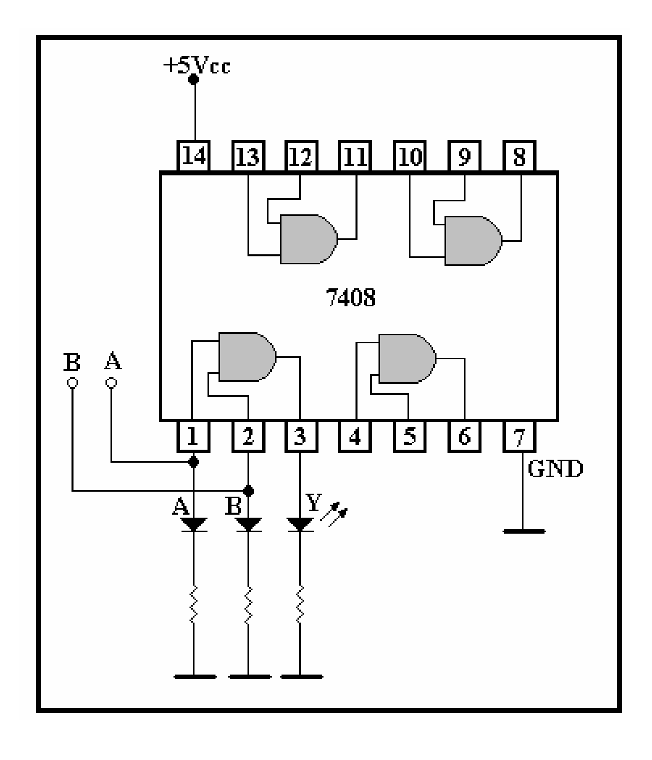
7404




**OBTENCIÓN DE LAS TABLAS DE VERDAD**

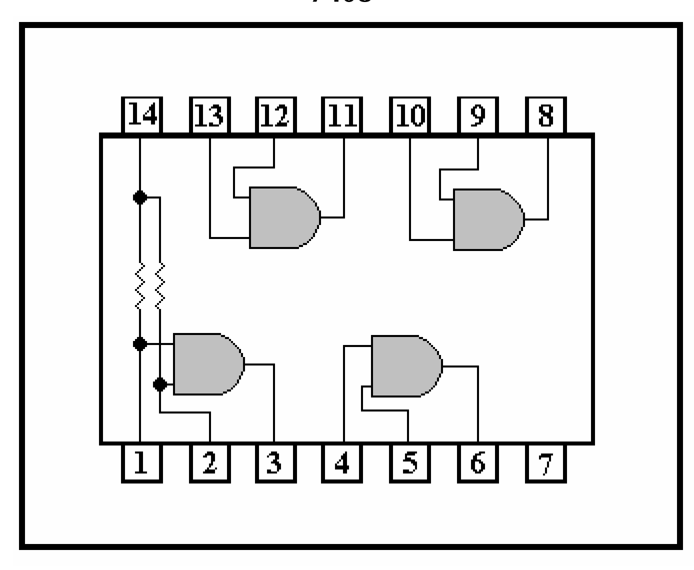
En el diagrama siguiente se muestra la configuración circuital en que se ha colocado al chip 7408, con el objeto de obtener la tabla de verdad de uno de sus operadores lógicos AND. Este chip contiene cuatro operadores lógicos AND de dos entradas cada uno, de los cuales en el diagrama de la figura se hace uso del que se encuentra conectado entre los terminales 1,2 y 3 del circuito integrado 7408. En cada uno de los terminales de entrada se encuentra conectado un diodo led, cuya finalidad es indicar visualmente el estado lógico de cada entrada. Otro led se encuentra conectado a la salida del operador a objeto de mostrar en todo momento la situación de la salida.

La tabla de la verdad se obtiene realizando con las entradas todas las posibles combinaciones, conectando a Vcc para obtener un "uno" y a masa para obtener un "cero".

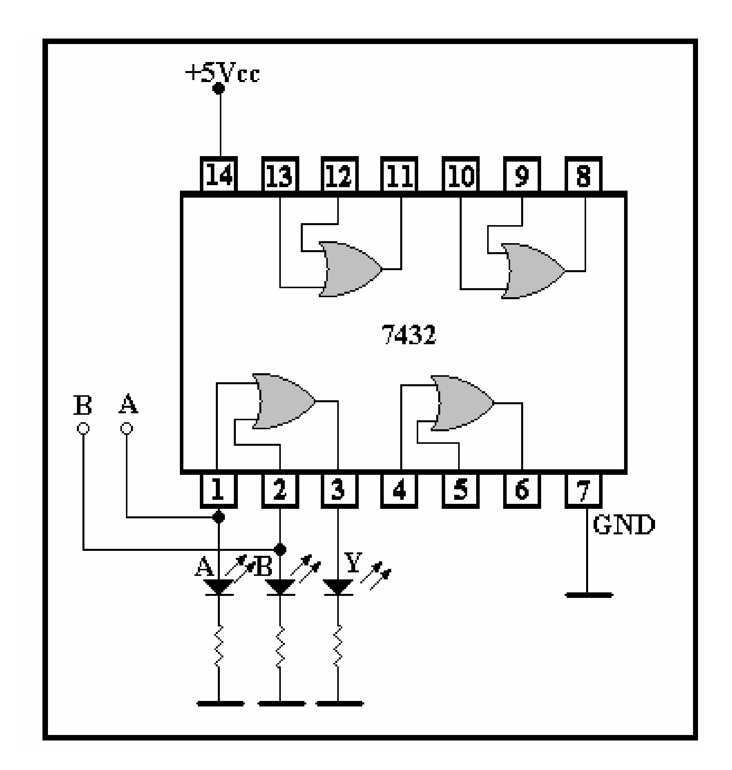


Notese que cuando todas las entradas quedan desconectadas (flotantes), el led de salida indica la existencia de un uno lógico en la salida, esto se debe a que cuando una entrada se encuentra flotante, esta queda conectada internamente a través de una resistencia a Vcc, tal como lo muestra la figura siguiente:

**7408**

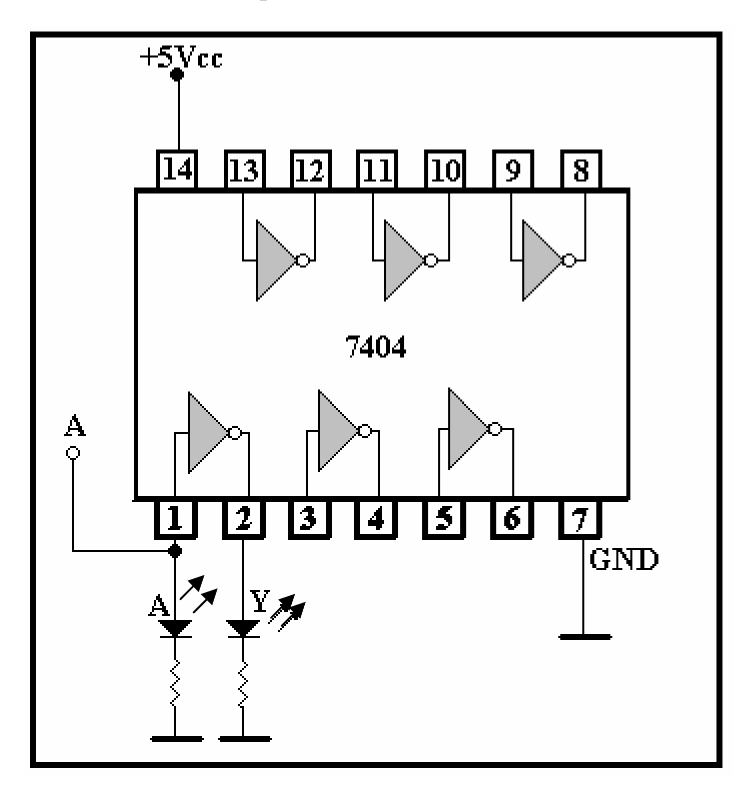



En la figura siguiente se muestra la disposición circuital para obtener la tabla de verdad del operador OR, para lo cual se utiliza ahora el chip 7432 el que contiene cuatro operadores OR de dos entradas cada uno.



Al igual que en el caso anterior, un led encendido indica un estado lógico "uno", mientras que un led apagado indica un estado lógico "cero".

Finalmente, en la próxima figura se muestra el circuito de prueba para obtener la tabla de verdad de un operador NOT, para lo cual se utiliza el chip 7404, el que contiene seis operadores NOT o inversores. En este caso al igual que en los anteriores, el resto de los operadores contenidos en el chip realizan exactamente la misma función.



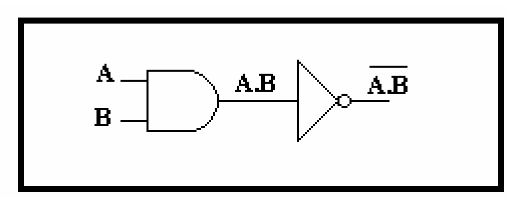

Hasta aquí hemos analizado la operación de los denominados circuitos básicos de lógica combinacional; como se ha podido observar, estos trabajan en función de solo dos estados eléctricos, los cuales se encuentran claramente diferenciados entre sí. El comportamiento de los circuitos lógicos estudiados, se ha representado a través de tablas de verdad, en las cuales se indican todos los posibles estados que puede tomar la salida, en función de las combinaciones a que se pueden realizar con los estados de entrada.

Combinando la acción de los circuitos lógicos estudiados, se obtienen los circuitos compuestos, en cuya operación se deberá también considerar la operatoria de los circui tos lógicos elementales, ya que básicamente las salidas de estos serán funciones obtenidas de funciones básicas; esto último será analizando en mayor detalle más adelante.

## COMPUERTA NAND

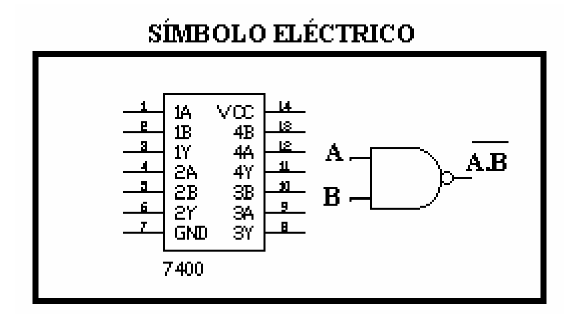
Como se muestra en la figura siguiente, este circuito se obtiene conectando un inversor a la salida de un circuito AND, en consecuencia, tal como se puede apreciar en la misma figura, la salida de este circuito corresponde al de la compuerta AND negada, ya que

entrega un cero a su salida solamente cuando todas sus entradas se encuentran en estado lógico “uno”, siendo la salida “uno” si una ó más entradas se encuentran en estado cero.

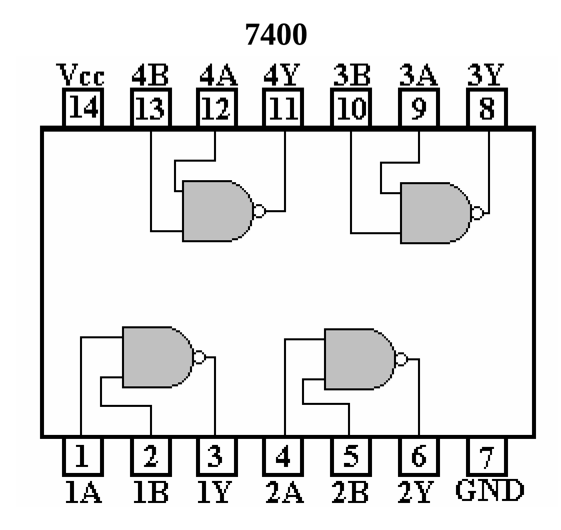



| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Normalmente el símbolo utilizado para representar esta compuerta es el mostrado en la figura siguiente:



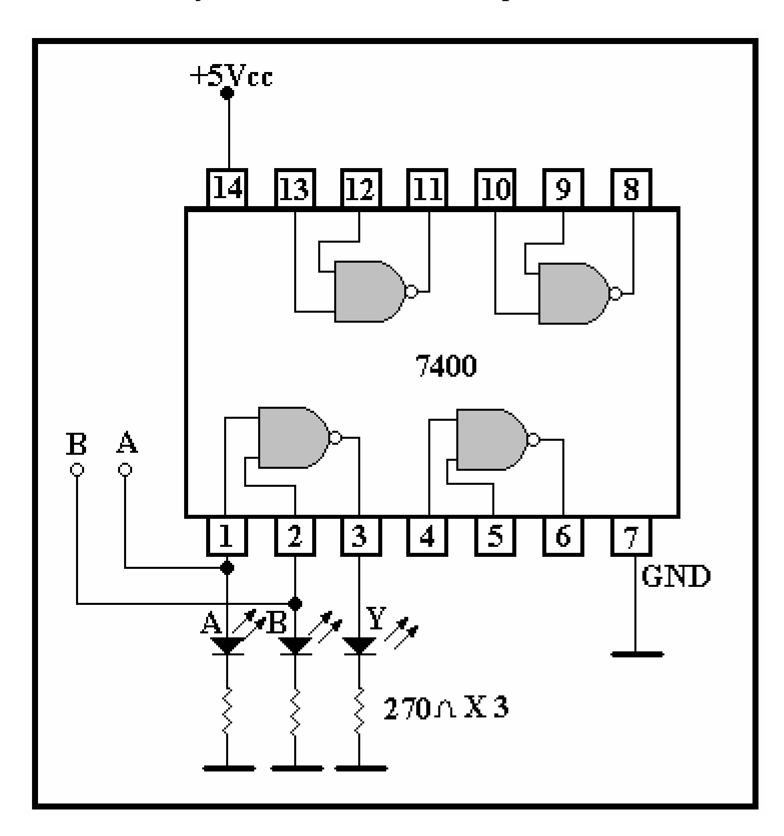

El circuito integrado que se muestra a continuación (IC 7400) contiene cuatro compuertas NAND, las cuales quedarán en condiciones de operar al aplicar polarización entre los terminales 7 y 14 del chip.





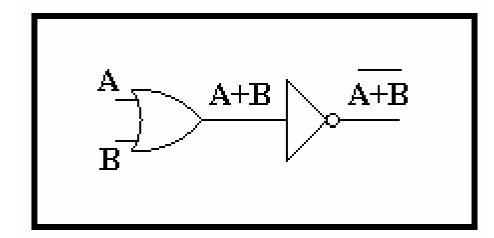



Para obtener la tabla de verdad del circuito en cuestión será necesario conectar una de las compuertas contenidas en el chip tal como lo muestra la figura siguien te. Luego mediante los terminales que han quedado libres (a y b), se realizarán las distintas combinaciones de entrada a objeto de obtener la correspondiente tabla de verdad.




## COMPUERTA NOR

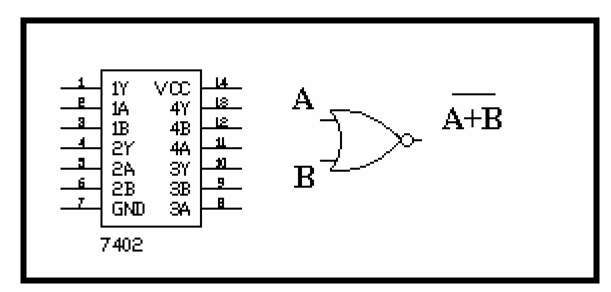
Este circuito se obtiene conectando a la salida de una compuerta OR un circuito inversor, lo cual se muestra en la figura siguiente:




| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

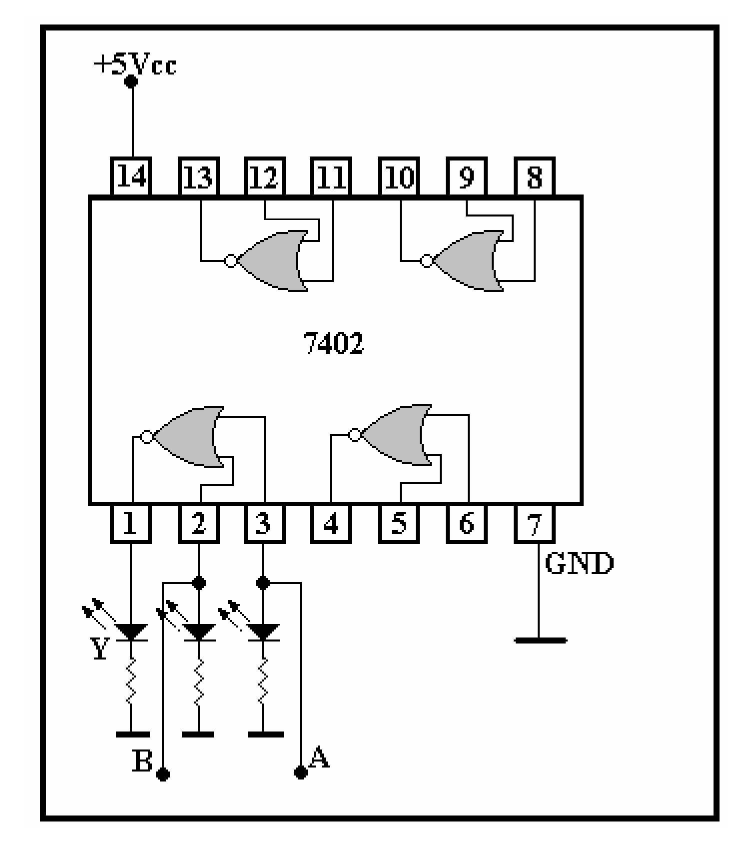
En esta figura se muestra además la función de la compuerta mediante la correspondiente tabla de la verdad. Básicamente la compuerta NOR entrega una función, la cual corresponde a la negación de la función realizada por la compuerta OR.

En la figura siguiente también se muestra el símbolo utilizado para representar a esta compuerta.



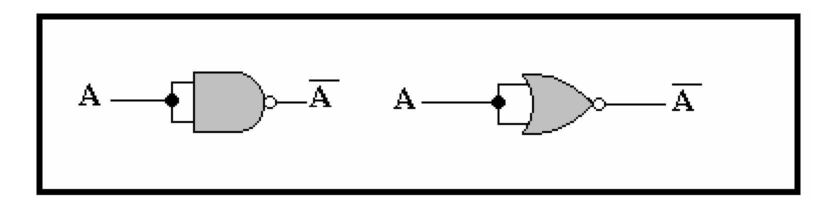

El circuito que permite obtener la tabla de verdad de este c ircuito se muestra en la próxima figura, donde se utiliza una compuerta que se encuentra contenida en el integrado 7402; al igual que en el caso anterior, las combinaciones se realizarán mediante los terminales indicados como “a” y “b”, correspondientes a las entradas del circuitos.

Es muy probable que en algunos casos, los estados de salida correspondientes a “uno lógico”, no resulten muy claros a través de los diodos led que se utilizan como indicadores, esto último se debe a que algunos circuitos integra dos no poseen la suficiente capacidad de corriente en sus salidas en estado lógico “uno”, por esta razón se recomienda utilizar algún tipo de indicador que produzca una mínima carga sobre el circuito bajo prueba.

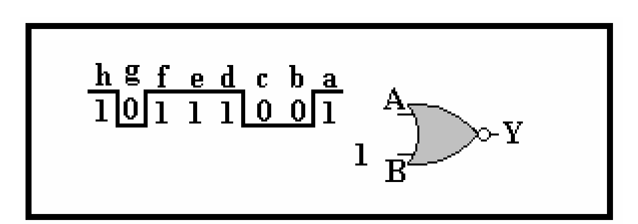


**NOTA:**

Cuando se puentean las entradas de una compuerta NAND o NOR, estas pasan a comportarse como una compuerta NOT o inversora.




**Ejercicios:** Indique cual seria el tren de pulsos de salida en la compuerta NOR si la entrada b se encuentra en 1 lógico:




**Solución:**

Pulso a =
Pulso b =
Pulso c =
Pulso d =
Pulso e =
Pulso f =
Pulso g =
Pulso h =

## COMPUERTA OR EXCLUSIVA (XOR)

Esta compuerta se caracteriza por entregar en su salida un bit “1” solo cuando en sus entradas hay un número impar de “1”, por esta razón se le considera como un detector de número impar de “1”.

El circuito lógico XOR puede ser formado con compuertas AND – OR e inversores.

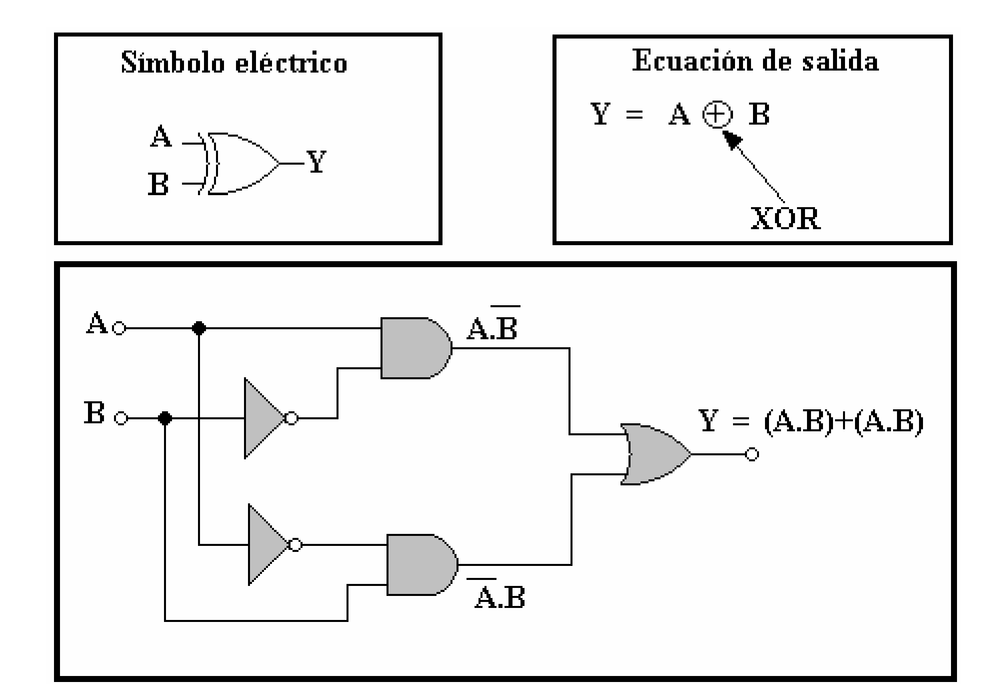
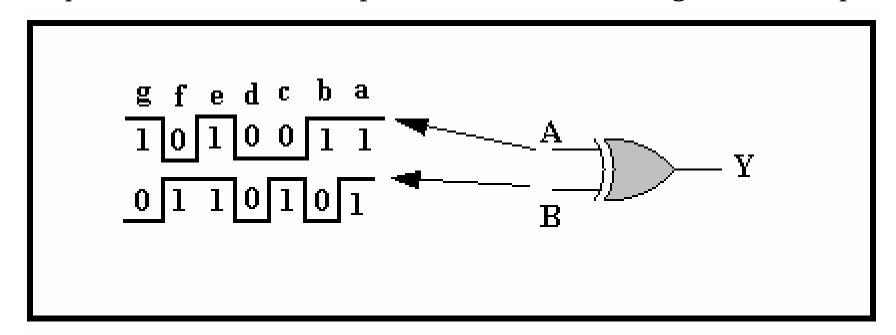


**Tabla de verdad**

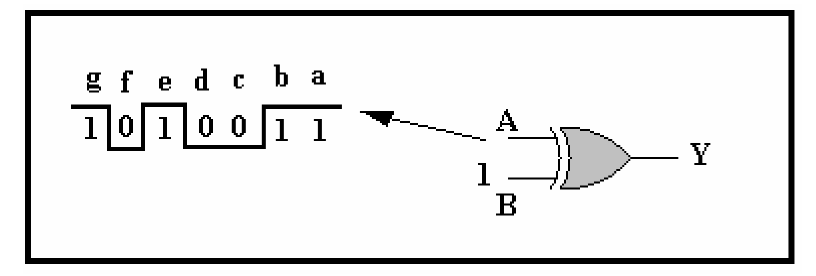
| ENTRADAS | | SALIDA |
|---|---|---|
| **B** | **A** | **Y** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Ejercicios: Indique cual será el tren de pulsos de salida de las siguientes compuertas:



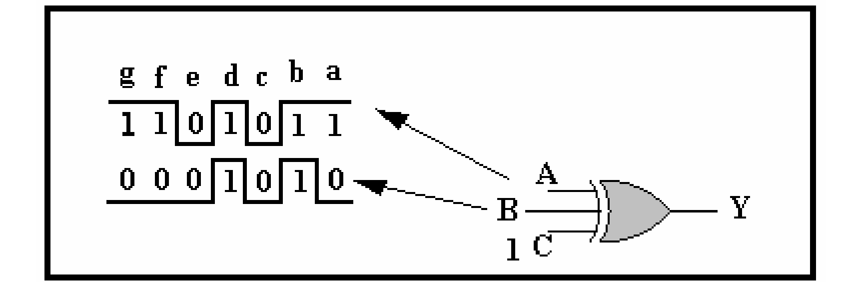

**Solución:**

Pulso a = Pulso e =
Pulso b = Pulso f =
Pulso c = Pulso g =
Pulso d = Pulso h =




**Solución:**

Pulso a = Pulso e =
Pulso b = Pulso f =
Pulso c = Pulso g =
Pulso d =



**Solución:**

Pulso a =
Pulso b =
Pulso c =
Pulso d =
Pulso e =
Pulso f =
Pulso g =

## COMPUERTA NOR EXCLUSIVA (XNOR)

Esta compuerta se caracteriza por negar las funciones XOR y por este motivo se considera a esta compuerta como un detector de número p ar de "1".

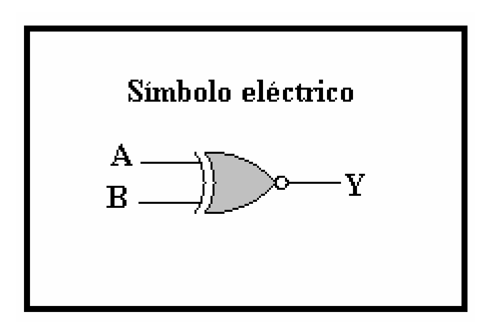
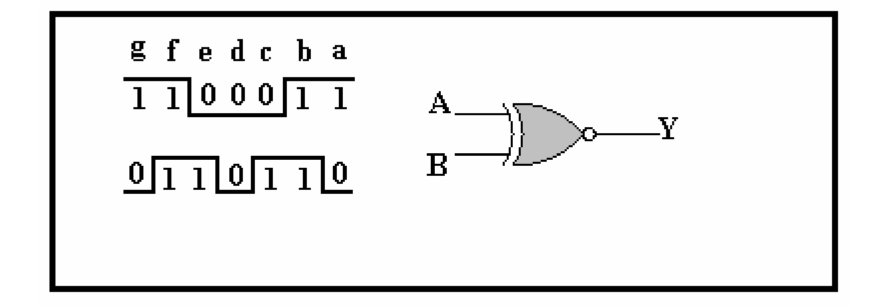



**Tabla de verdad para las compuertas XOR y XNOR**

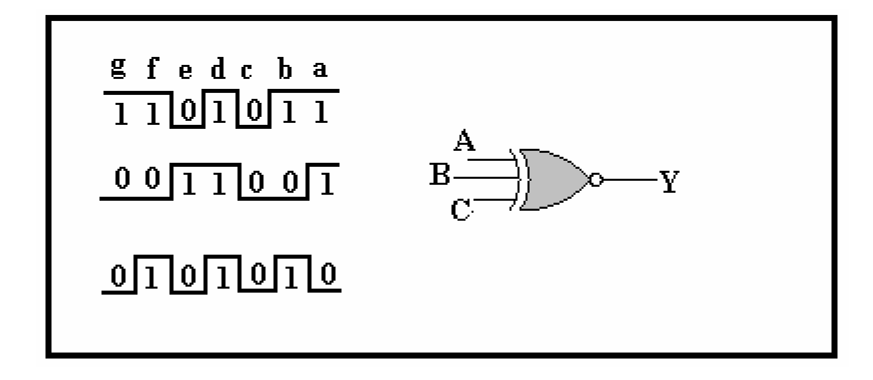
| ENTRADAS | | SALIDAS | |
|---|---|---|---|
| **B** | **A** | **XOR** | **XNOR** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Ejercicios:** Indique cual será el tren de pulsos de salida de las siguientes compuertas:



**Solución:**

Pulso a = Pulso e =
Pulso b = Pulso f =
Pulso c = Pulso g =
Pulso d =

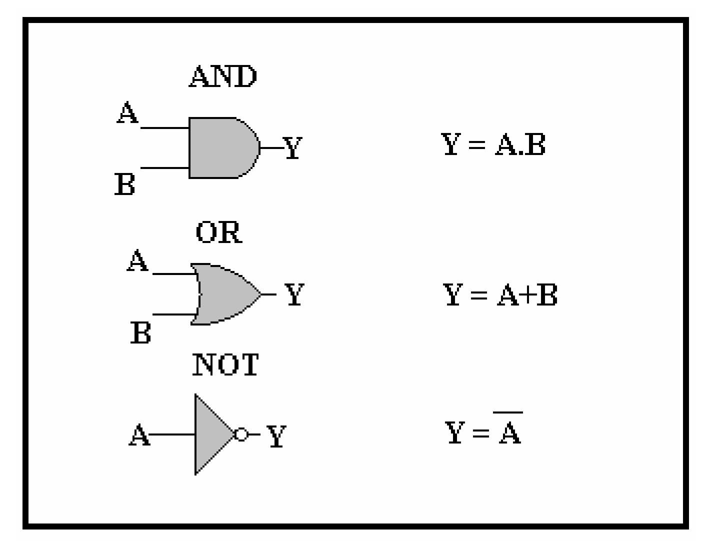



**Solución:**

Pulso a = Pulso e =
Pulso b = Pulso f =
Pulso c = Pulso g =
Pulso d =

## EXPRESION LITERAL DE LAS FUNCIONES LOGICAS

Con el objeto de simplificar tanto la representación como el procesamiento de las funciones lógicas, es que estas se representan en forma literal de acuerso a lo indicado en la figura siguiente:



De acuerdo a lo indicado, la función **AND** se representa relacionando la variables de entrada con el símbolo "**.**" o bien, " ", tal como se muestra a continuación:

**Y =A.B.C.D o bien Y = ABCD**

Esto se lee "la salida y es función de las variables de entrada A,B, C y D", lo cual significa que la salida del circuito **AND** depende en todo momento del estado en que se encuentren las variables de entrada identificadas como A, B, C, y D.

La función **OR** se representa relacionando las variables de entrada con el símbolo "+", en consecuencia, toda vez que aparezca el signo MAS entra las vari ables de un sistema, significa que estas se encuentran conectadas a la entrada de una compuerta **OR**.

Así, entonces, la función **OR** se representara literalmente tal como se indica a continuación:

**Y = A + B + C + D**

Tanto en la función **AND** como en la función **OR**, el número de variables de entrada puede ser cualquiera, lo cual significa que podremos tener tablas de verdad con muchas combinaciones.

Finalmente la función **NOT**, se representa mediante una línea sobre la variable negada o también mediante una comilla, tal como se muestra a continuación:

**Y = A = Á o bien Y = A =$\overline{\text{Ā}}$**

Todo circuito de lógica combinacional se puede representar a través de una expresión literal, así como también es posible transformar una expresión literal a un circuito de lógica combinacional.

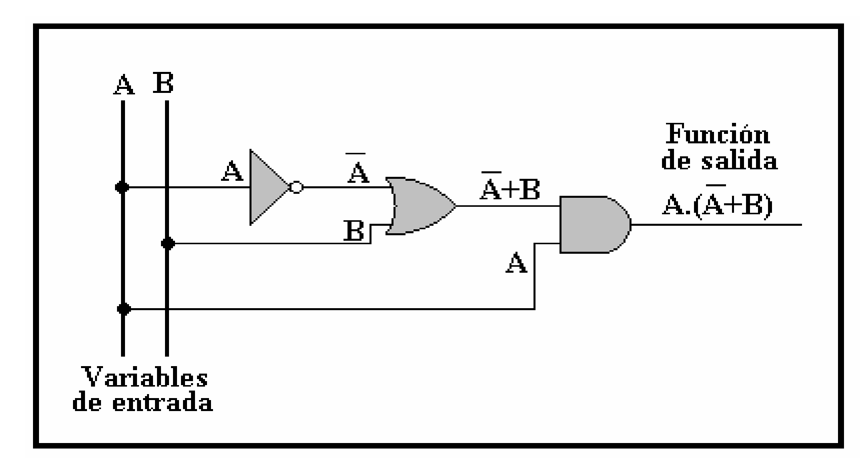
Para transformar una expresión literal a un circuito de lógica, es necesario seguir el siguiente orden de prioridad de las operaciones lógicas:

**1) Resolver paréntesis**
**2) Realizar operaciones AND**
**3) Realizar operaciones OR**

Los complementos se resolverán en el orden que corre sponda para ejecutar las operaciones antes indicadas.

En la figura siguiente se muestra cómo se transformó a un circuito lógico la expresión literal:

**Y = A . (Ā + B)**




Nótese que en la implementación de este circuito se ha utilizado una compuerta **OR** de dos entradas, una compuerta **AND** también de dos entradas y un inversor **NOT**.
En este diagrama las dos lineas verticales son los conductore s que poseen el estado de las variables A y B. Luego, siguiendo el orden de prioridad de las operaciones lógicas se resuelve el paréntesis, de lo cual resulta una compuerta **OR** de dos entradas a las cuales deben llegar las variables A y B. Nótese que la variable "A" debe llegar complementada, razón por la cual se utiliza un inversor.
En estas condiciones a la salida de la compuerta **OR** se tendrá una función lógica que tendrá la siguiente expresión literal:

$$\bar{A} + B$$

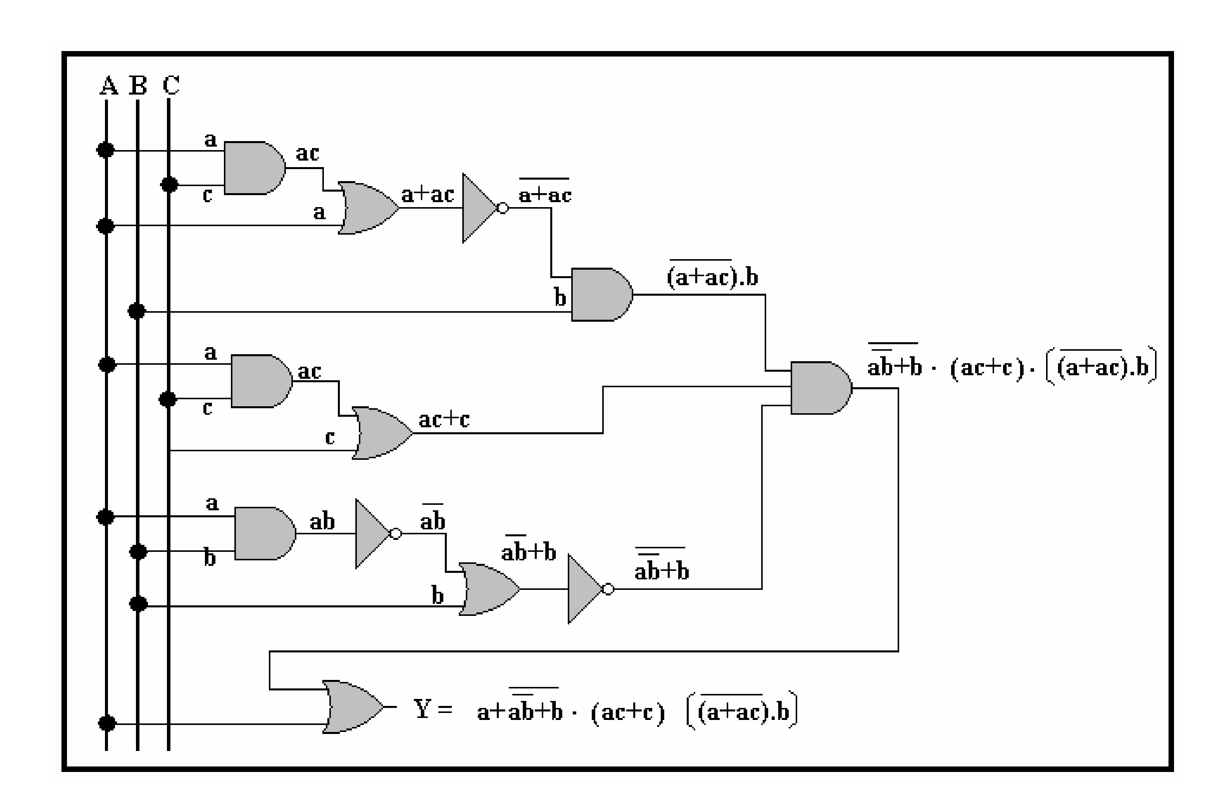
De acuerdo con la expresión literal de la función, la salida de la compuerta **OR** debe ingresar a la compuerta **AND** de dos entradas, debiendo ingresar por la segunda entrada la misma variable "A", pero ahora sin complementar. De acuerdo con lo anterio rmente señalado, a la salida de la compuerta **AND** se obtiene la operatoria correspondiente entre lo que ingresó por la entrada superior (salida de la compuerta **OR**) y lo que ingreso por la entrada inferior (variable "A"). En la figura se muestra además, en cada parte del circuito, la expresión literal que corresponde a la función presente en ese punto.

A través de este ejemplo propuesto queda de manifiesto que para implementar el circuito a partir de una expresión literal, es necesario descomponer a esta úl tima en subfunciones que se puedan implementar con los circuitos y compuertas conocidos.

Normalmente la cantidad de compuertas utilizadas en la implementación de un circuito de lógica, así como el número de entradas, que posean las mismas, es proporcional a la extensión de la expresión literal que le corresponde, aunque en algunos casos como veremos más adelante, es posible encontrar una expresión literal más reducida a objeto de implementar un circuito también más reducido y que realice la misma función.

La figura siguiente muestra la transformación a circuito de la función cuya expresión literal es:

$$Y = A + \overline{\overline{AB} + B} \; . \; (AC + C) \left[ \overline{(A + AC)}\, B \right]$$



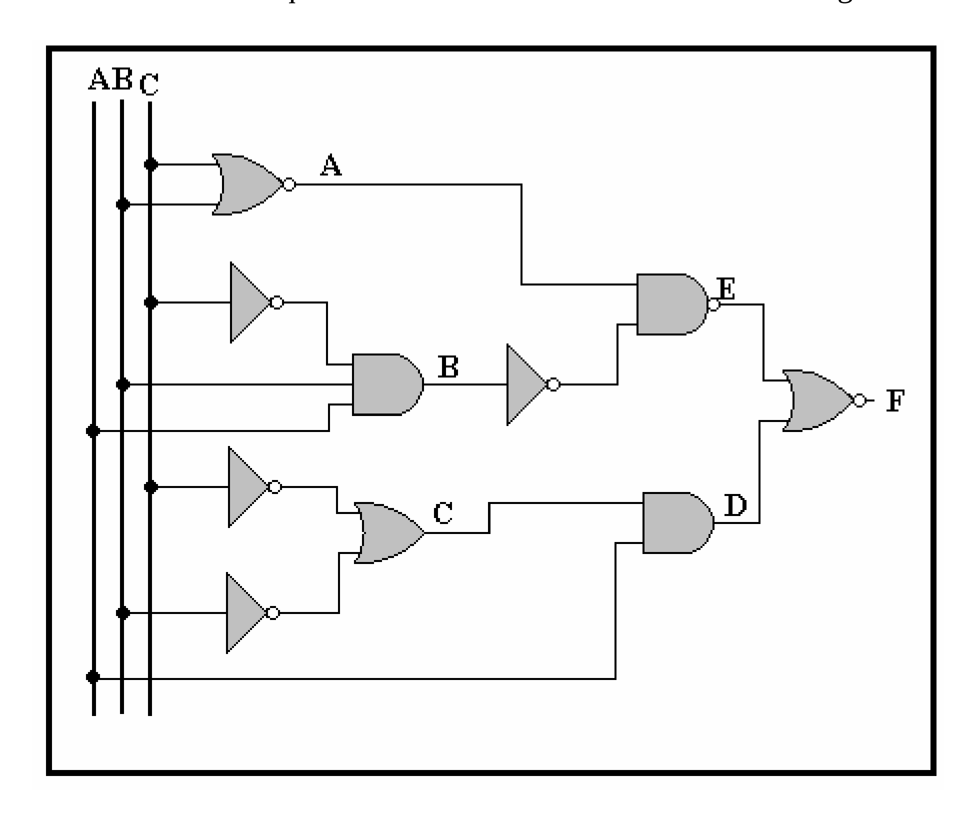

Nótese que en esta expresión literal existen operaciones lógicas entre paréntesis que a su vez, se encuentran entre paréntesis más externos; en esto s casos se comenzará primero por resolver aquellos más externos para continuar hacia fuera con los siguientes. En el diagrama mostrado en la figura se muestra en los distintos puntos del circuito la expresión literal parcial que le corresponde, por lo tan to al avanzar hacia la salida del circuito se obtiene la expresión literal original.

**Ejercicios:** Transformar en circuito las siguientes expresiones literales:

**Y = a + b + ca**
**Y = (ab + c) ac + b**
**Y = abc + bd + bc**
**Y = a (cd + ab) (((a + b) c + d) a + ad)**
**Y = bc + bd + ab + ac**

Así como se puede transformar una expresión literal a un circuito de lógica combinacional, también es posible transformar un circuito a una expresión literal. Para obtener la expresión literal de un circuito de lógica combinacional, se coloca en la salida de cada compuerta la operación lógica que le corresponde con las variables que posee como entradas, así, al avanzar hacia la salida del circuito se va construyendo la expresión literal de la función que le corresponde.

En la figura siguiente se muestra como obtener la expresión literal de una función que se encuentra implementada en un circuito. Los puntos indicados desde A hasta E corresponden a subfunciones que actúan como entradas de los circuitos siguientes




$A = \overline{b+c}$

$B = a.b.\overline{c}$

$C = \overline{b}+c$

$D = a.c = a.(\overline{b}+\overline{c})$

$E = \overline{a+\overline{b}} = \overline{\overline{b+c} \,.\, \overline{a \,.\, b\overline{c}}}$

$F = e+d = \overline{\overline{a+b}+d}$

**Expresión resultante**

$$Y = \overline{\overline{\overline{b+c} \;.\; \overline{ab\overline{c}}} + (\overline{b}+\overline{c}) \;.\; a}$$

## CONVERSIÓN DE COMPUERTAS USANDO INVERSIONES

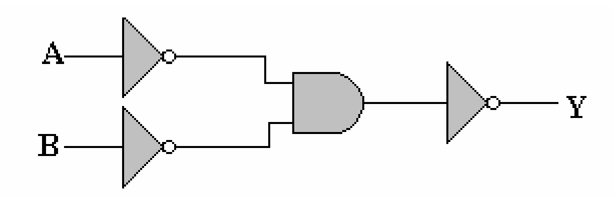
Cuando se usan compuertas lógicas, surge la necesidad de convertir a otras funciones lógicas. Un método de conversión sencillo es el de colocar inversores en las entradas y/o salidas de las compuertas.

**Ejemplo:**

| INVERSORES EN LA ENTRADA | COMPUERTA ORIGINAL | INVERSOR EN LA SALIDA | NUEVA FUNCIÓN LÓGICA |
|---|---|---|---|
| | AND | NOT | OR |
| | OR | NOT | AND |
| | NAND | NOT | NOR |
| | NOR | NOT | NAND |

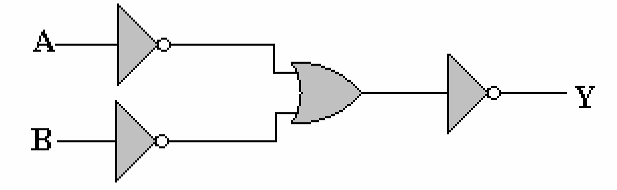
| COMPUERTA ORIGINAL | INVERSOR AGREGADO | NUEVA FUNCIÓN LÓGICA |
|---|---|---|
| AND | NOT | NAND |
| NAND | NOT | AND |
| OR | NOT | NOR |
| NOR | NOT | OR |

Ejercicios: Desarrolle la tabla de verdad, ecuación de salida y compuerta equivalente de los siguientes circuitos lógicos:



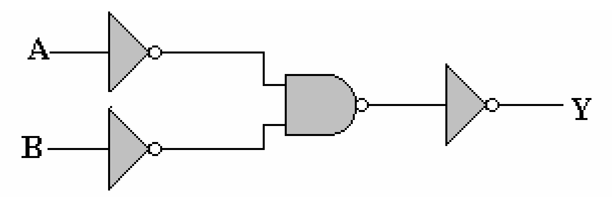

**Circuito equivalente a una compuerta: ....................**

| ENTRADAS | | SALIDA |
|---|---|---|
| **B** | **A** | **Y** |
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |



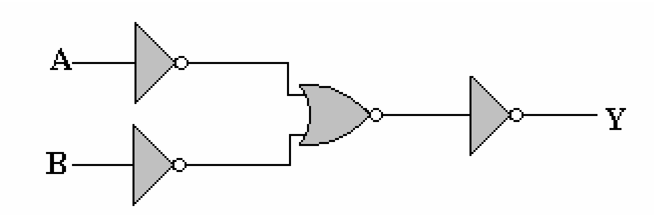

**Circuito equivalente a una compuerta: ....................**

| ENTRADAS | | SALIDA |
|---|---|---|
| B | A | Y |
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |



Circuito equivalente a una compuerta: ....................

| ENTRADAS | | SALIDA |
|---|---|---|
| B | A | Y |
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |



Circuito equivalente a una compuerta: ....................

| ENTRADAS | | SALIDA |
|---|---|---|
| B | A | Y |
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

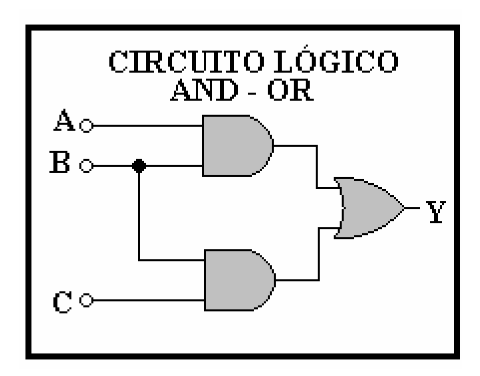
Señale cual sería la compuerta resultante si en los cuatro ejemplo s señalados se eliminara el inversor de salida.

1) Compuerta _______________________
2) Compuerta _______________________
3) Compuerta _______________________
4) Compuerta _______________________

## COMBINACIÓN DE COMPUERTAS LOGICAS

Muchos problemas cotidianos de lógica digital emplean diversas compuertas lógicas. El patrón más común de compuertas es el AND – OR.

1)



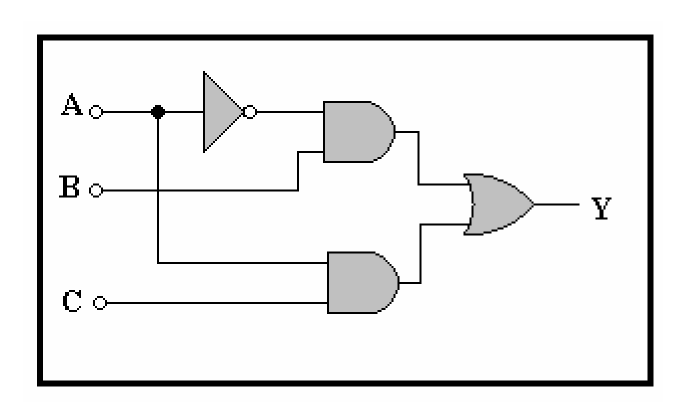

| ENTRADAS | | | SALIDA |
|---|---|---|---|
| **C** | **B** | **A** | **S** |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Expresión Booleana**

**Y = A . B + B . C** **o bien** **Y = AB + BC**

Los circuitos lógicos **AND – OR** pueden presentar diversas variantes cuando funcionan en combinación con compuertas inversoras. A continuación se presentan diversas combinaciones a las cuales se deberá desarrollar su respectiva tabla de verdad, basándose en los ejemplos antes enunciados, agregando además la correspondiente expresión booleana:

2)

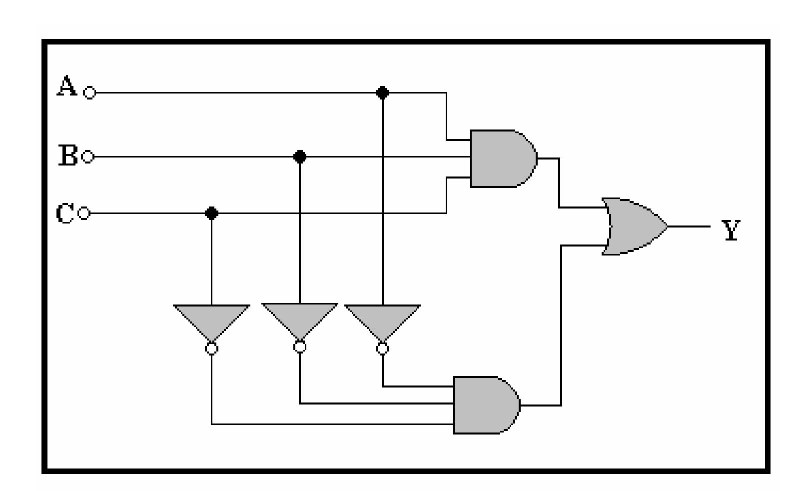



| ENTRADAS | | | SALIDA |
|---|---|---|---|
| **C** | **B** | **A** | **S** |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Expresión booleana:**

**Y** =

3)

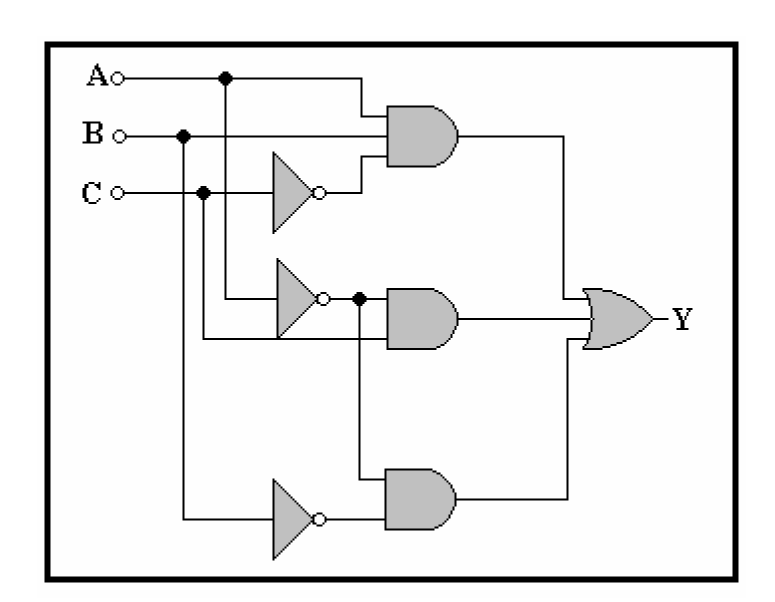


| ENTRADAS | | | SALIDA |
|---|---|---|---|
| **C** | **B** | **A** | **S** |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Expresión booleana:**

**Y** =

4)

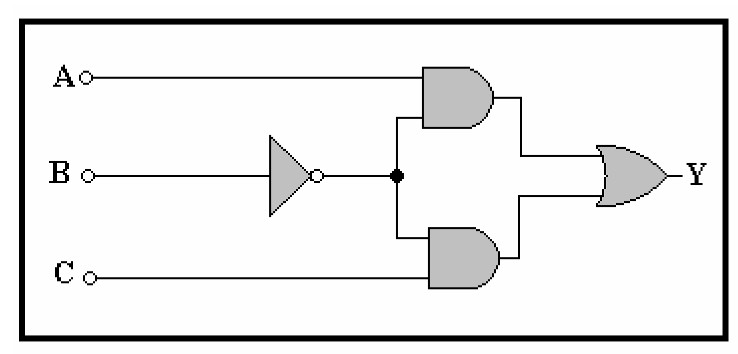



| ENTRADAS | | | SALIDA |
|---|---|---|---|
| **C** | **B** | **A** | **S** |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Expresión booleana:**

**Y** =

5)

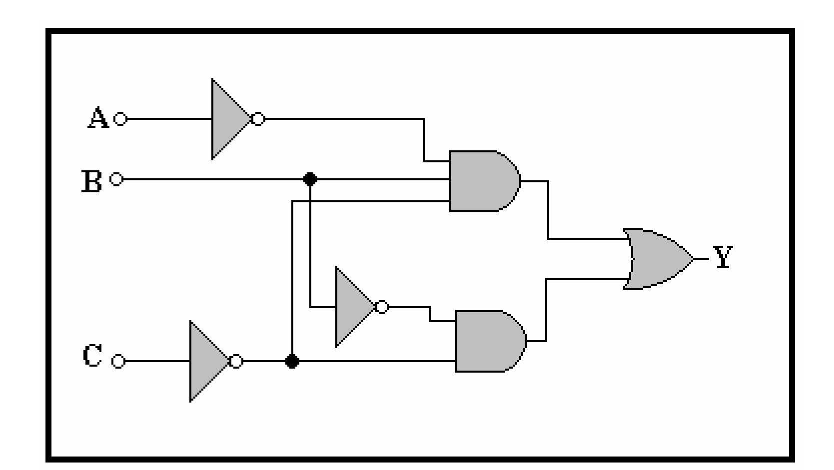



| ENTRADAS | | | SALIDA |
|---|---|---|---|
| **C** | **B** | **A** | **S** |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Expresión booleana:**

**Y** =

**6)**

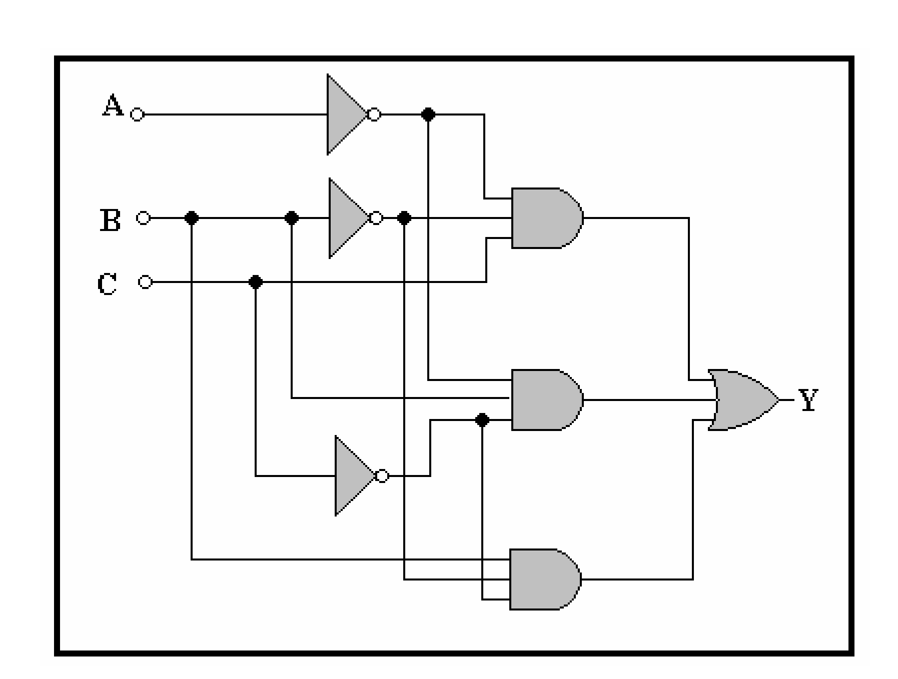


| ENTRADAS | | | SALIDA |
|---|---|---|---|
| **C** | **B** | **A** | **S** |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Expresión booleana:**

**Y =**

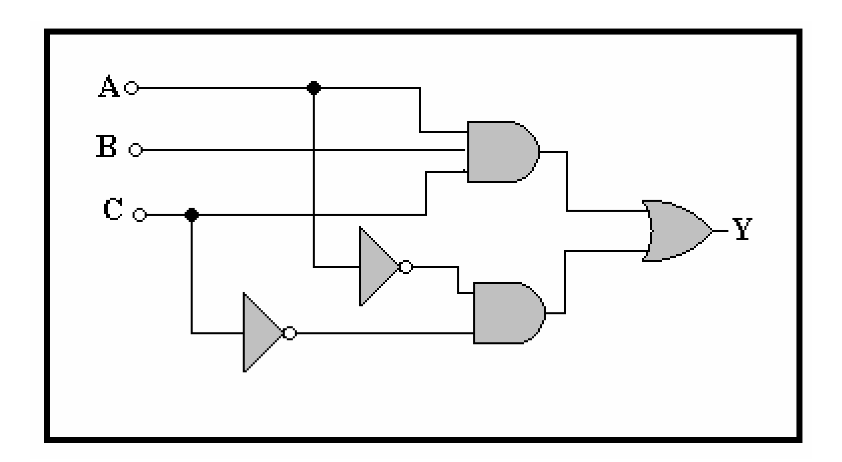
7)




| ENTRADAS | | | SALIDA |
|---|---|---|---|
| **C** | **B** | **A** | **S** |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Expresión booleana:**

**Y =**



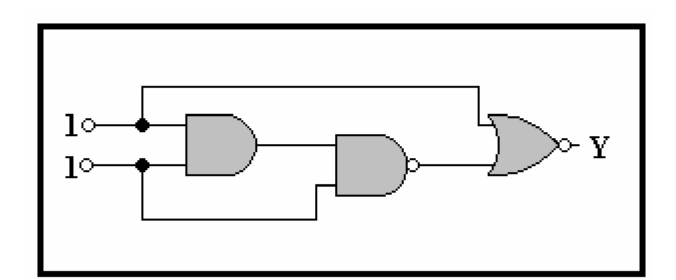

| ENTRADAS | | | SALIDA |
|---|---|---|---|
| **C** | **B** | **A** | **S** |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Expresión booleana:**

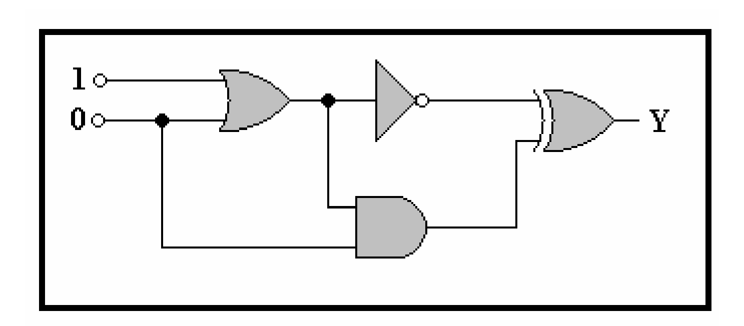
**Y =**

Señale el nivel de salida de los siguientes circuitos lógicos (0 ó 1):

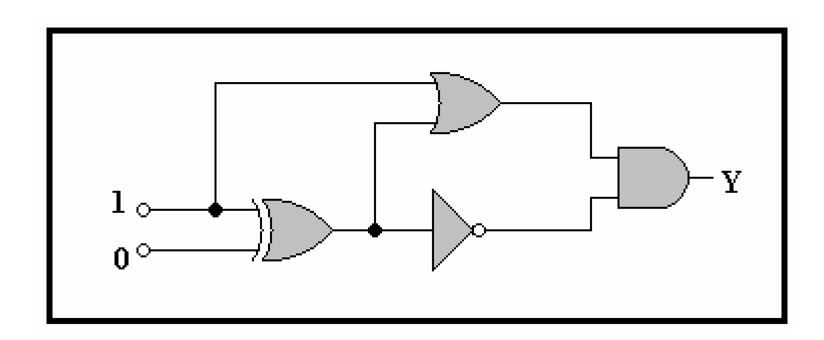
1.-



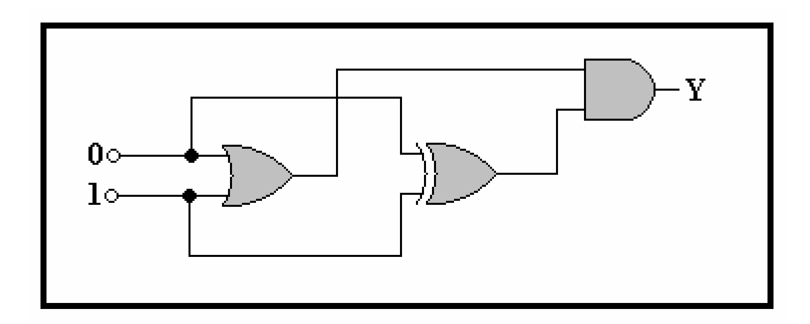
2.-



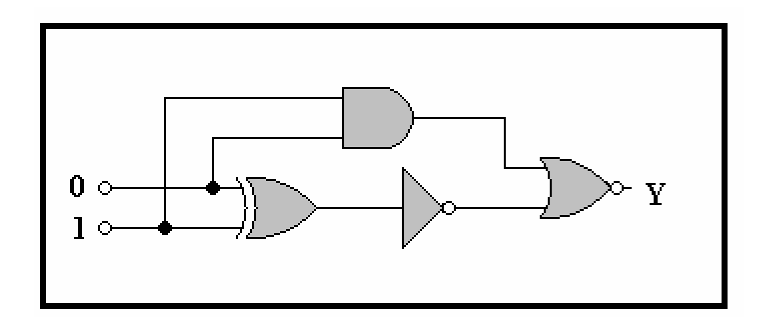

3.-



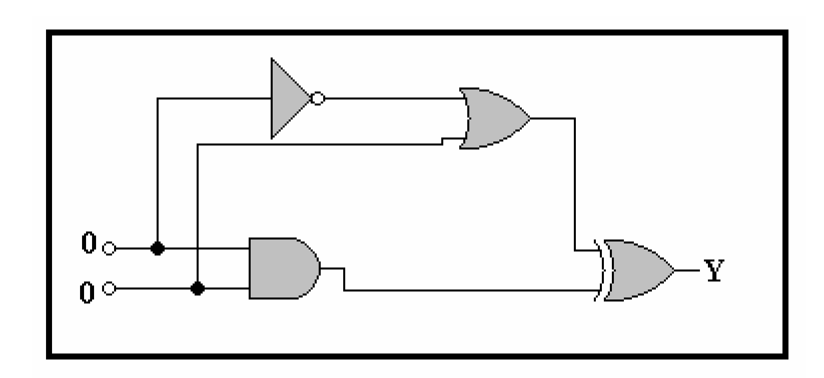

4.-



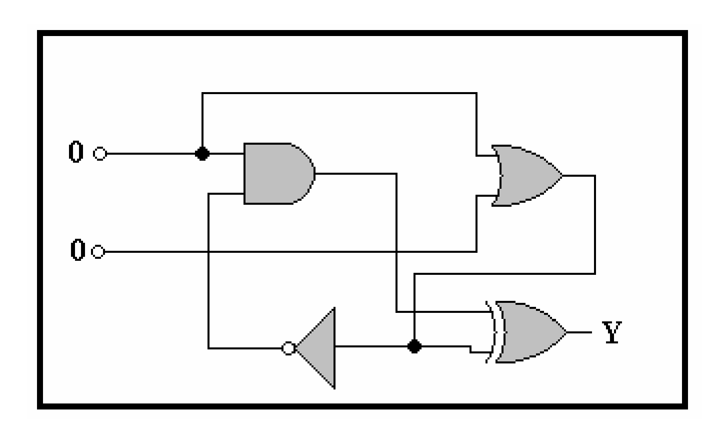

5.-



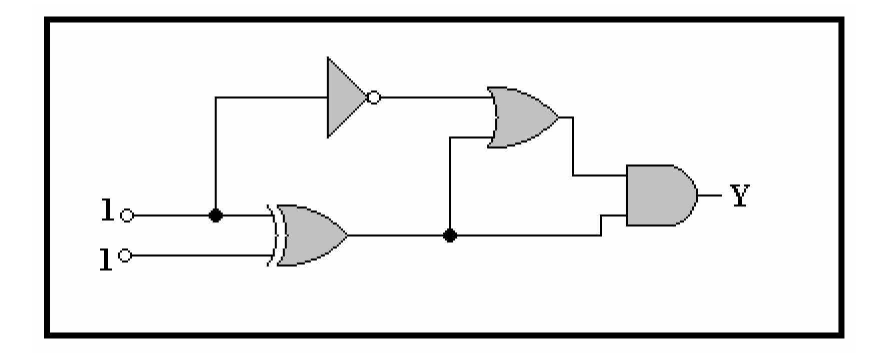
6.-



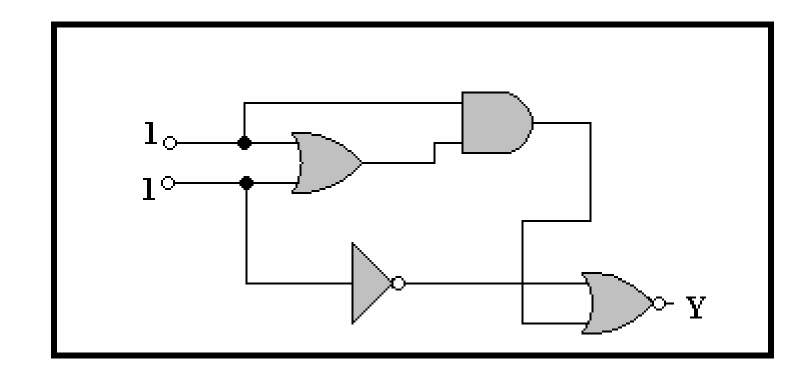

7.-



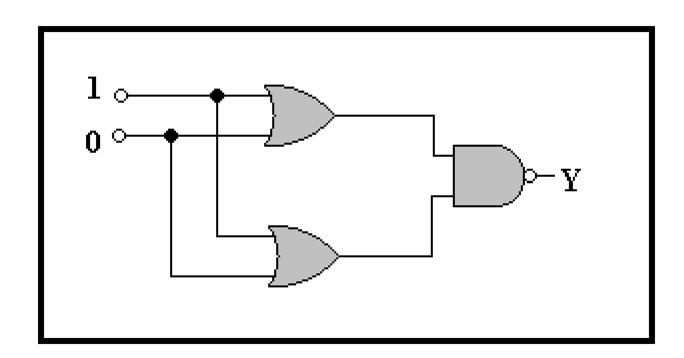

8.-




9.-



10.-




**Salida de los circuitos lógicos:**

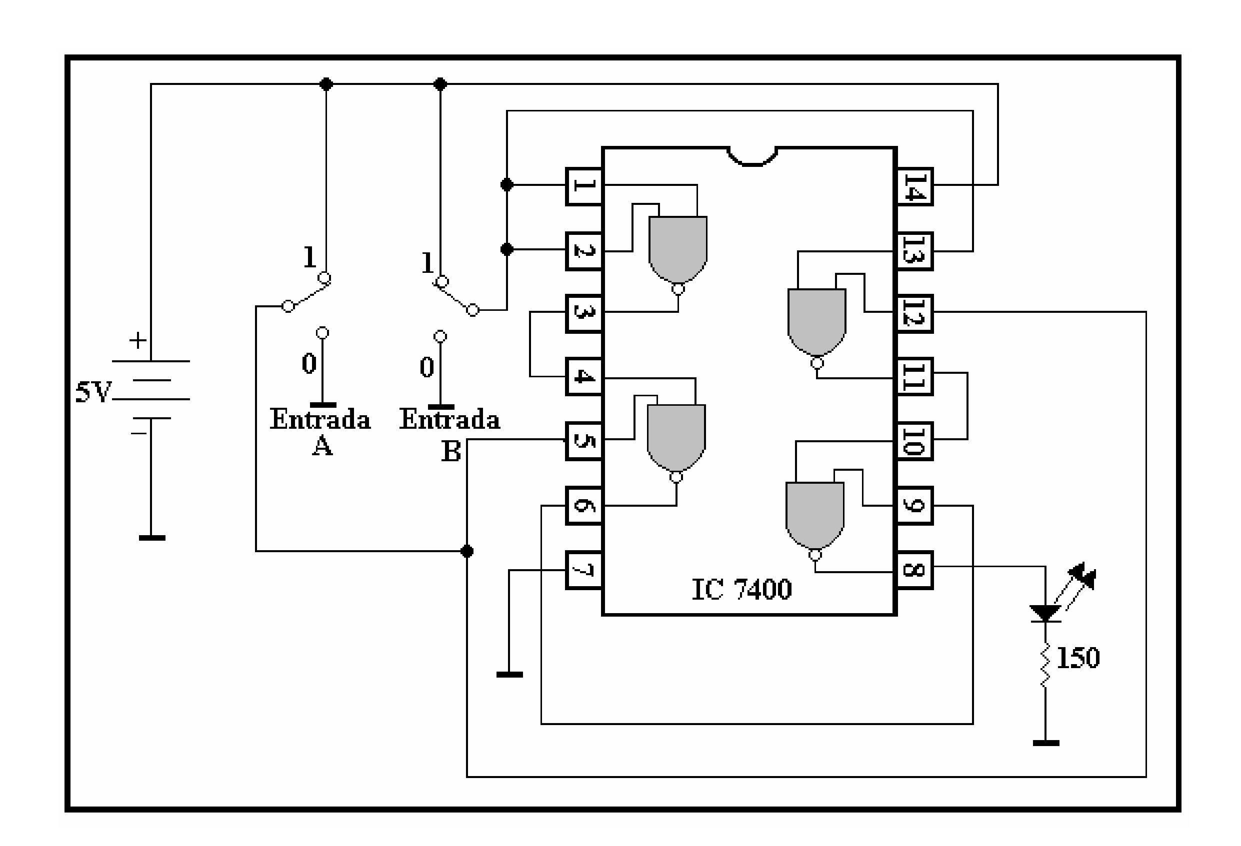
1) Y =__________________ 6) Y =__________________
2) Y =__________________ 7) Y =__________________
3) Y =__________________ 8) Y =__________________
4) Y =__________________ 9) Y =__________________
5) Y =__________________ 10) Y =_________________

**Ejercicio: Sobre el circuito lógico representado a continuación (IC – 7400), realice el siguiente trabajo:**

a) Dibuje el conexionado de las compuertas NAND, partiendo de las entradas A y B hasta finalizar en el diodo led.

b) Dibuje la tabla de verdad del circuito.

c) Señale si el diodo led quedará encendido o apagado cuando las entradas A y B se encuentren en Nivel 1.




## CODIGOS BINARIOS

Los sistemas digitales sólo procesan números binarios (ceros y unos) , de los cuales han surgido varios códigos para realizar funciones específicos en equipos digitales. Estos códigos usan ceros y unos, pero sus significados pueden variar según el código.

Para pasar de código a código en los sistemas digitales se emplean los traductores electrónicos, también denominados codificadores y decodificadores.

**CODIGOS BINARIOS PESADOS**

Código decimal codificado en Binario (BCD)

Bcd = Del Inglés Binary Coded Decimal.

| BCD | | | | |
|---|---|---|---|---|
| **DECIMAL** | **8** | **4** | **2** | **1** |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |

El nombre más difundido de este código es el de BCD 8421. El 8421 se refiere al peso que se le da a cada lugar en el código de cuatro Bit.

El Bit más significativo tiene un peso de 8 mientras que el menos significativo sólo tiene peso de 1.

Existen varios códigos BCD que tienen otros pesos para los cuatro lugares.

Como el código BCD 8421 es el más popular, se acostumbra a referirse a él simplemente como el Código BCD.

**CONVERSIÓN DECIMAL A BCD**

| | | | |
|---|---|---|---|
| **Decimal** | 1 | 5 | 0 |
| | ↓ | ↓ | ↓ |
| **BCD** | 0001 | 0101 | 0000 |

**CONVERSIÓN BCD A DECIMAL**

| | | |
|---|---|---|
| **BCD** | 1001 | 0110 |
| | ↓ | ↓ |
| **Decimal** | 9 | 6 |

**CONVERSIÓN DECIMAL FRACCIONARIO A BCD**

| | | | | | |
|---|---|---|---|---|---|
| **Decimal** | 3 | 2 | | 8 | 4 |
| | ↓ | ↓ | | ↓ | ↓ |
| **BCD** | 0011 | 0010 | , | 1000 | 0100 |







## CONVERSIÓN BCD FRACCIONARIO A DECIMAL

| | | | | | |
|---|---|---|---|---|---|
| **BCD** | 0111 | 0001 | , | 0000 | 1000 |
| | ↓ | ↓ | | ↓ | ↓ |
| **DECIMAL** | 7 | 1 | , | 0 | 8 |

Para realizar esta última conversión, primero se debe dividir el número BCD en grupos de cuatro, cada uno a partir del punto Binario, luego cada grupo de 4 se traduce al número decimal correspondiente y el punto Binario se transforma en el punto decimal.

**Ejercicios: Convierta los siguientes números BCD 8421 a sus equivalentes decimales:**

a) 1010 = b) 00010111 =
c) 10000110 = d) 010101000011 =

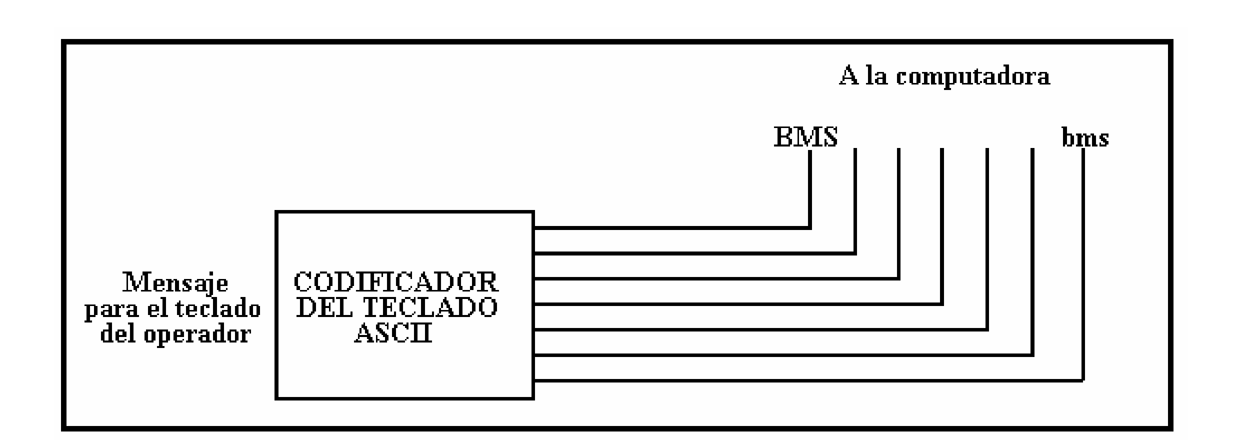
## CODIGOS ALFANUMERICOS

| **CARÁCTER** | ASC II | EBCDIC | CARACTER | ASC II | EBCDIC |
|---|---|---|---|---|---|
| **Espacio** | 010 0000 | 0100 0000 | A | 100 0001 | 1100 0001 |
| **!** | 010 0001 | 0101 1010 | B | 100 0010 | 1100 0010 |
| **“** | 010 0010 | 0111 1111 | C | 100 0011 | 1100 0011 |
| **#** | 010 0011 | 0111 1011 | D | 100 0100 | 1100 0100 |
| **$** | 010 0100 | 0101 1011 | E | 100 0101 | 1100 0101 |
| **%** | 010 0101 | 0110 1100 | F | 100 0110 | 1100 0110 |
| **&** | 010 0110 | 0101 0000 | G | 100 0111 | 1100 0111 |
| **‘** | 010 0111 | 0111 1101 | H | 100 1000 | 1100 1000 |
| **(** | 010 1000 | 0100 1101 | I | 100 1001 | 1100 1001 |
| **)** | 010 1001 | 0101 1101 | J | 100 1101 | 1101 0001 |
| ***** | 010 1010 | 0101 1100 | K | 100 1011 | 1101 0010 |
| **+** | 010 1011 | 0100 1110 | L | 100 1100 | 1101 0011 |
| **,** | 010 1100 | 0110 1011 | M | 100 1101 | 1101 0100 |
| **-** | 010 1101 | 0110 0000 | N | 100 1110 | 1101 0101 |
| **.** | 010 1110 | 0100 1011 | O | 100 1111 | 1101 0110 |
| **/** | 010 1111 | 0110 0001 | P | 101 0000 | 1101 0111 |
| **0** | 011 0000 | 1111 0000 | Q | 101 0001 | 1101 1000 |
| **1** | 011 0001 | 1111 0001 | R | 101 0010 | 1101 1001 |
| **2** | 011 0010 | 1111 0010 | S | 101 0011 | 1110 0010 |
| **3** | 011 0011 | 1111 0011 | T | 101 0100 | 1110 0011 |
| **4** | 011 0100 | 1111 0100 | U | 101 0101 | 1110 0100 |
| **5** | 011 0101 | 1111 0101 | V | 101 0110 | 1110 0101 |
| **6** | 011 0110 | 1111 0110 | W | 101 0111 | 1110 0110 |
| **7** | 011 0111 | 1111 0111 | X | 101 1000 | 1110 0111 |
| **8** | 011 1000 | 1111 1000 | Y | 101 1001 | 1110 1000 |
| **9** | 011 1001 | 1111 1001 | Z | 101 1010 | 1110 1001 |

Los Bit pueden ser codificados para representar números, letras del alfabeto y signos de puntuación, dando origen a los códigos alfanuméricos. Uno de estos códigos (7 Bit) es el **AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII,** se pronuncia **"ASKI")** este código se utiliza ampliamente en las computadoras pequeñas, para traducir de los caracteres del teclado al lenguaje de la computadora. La tabla de la Figura anterior muestra una lista completa de todas las combinaciones del **Código ASCII.**

Otro código ampliamente utilizado es el **EXTENDED BINARY – CODED – DECIMAL INTERCHANGE CODE (EBCDIC,** se pronuncia **"EBSIDIK").** Una parte del código **EBCIDIC** se muestra en la Figura anterior. Se puede advertir que utiliza 8 Bitz y por lo tanto puede tener más variaciones y caracteres que el código **ASCII**; por esta razón el código **EBCDIC** se utiliza en muchas de las computadoras más grandes.

## SISTEMA DEL CODIFICADOR DEL ASCII




**PROBLEMA: Codifique en codigo ascii la palabra 7 electrones**

| | |
|---|---|
| 7 = | T = |
| ESPACIO = | R = |
| E = | O = |
| L = | N = |
| E = | E = |
| C = | S = |

## CONVERSIÓN DE CODIGOS

Una de las aplicaciones que tienen las compuertas lógicas en los sistemas digitales es la de convertidores de códigos, de los cuales los más utilizados son el BINARIO, BCD (8421), OCTAL, HEXADECIMAL y por supuesto el DECIMAL.

La principal ventaja de los convertidores de código es que permiten convertir el lenguaje humano (decimal) a lenguaje de máquina y viceversa.

## DIAGRAMA BASICO DE UNA CALCULADORA

En la figura representada el sistema de entrada de datos es el conjunto de teclas. Entre este conjunto y la unidad central de proceso (CPU) se encuentra un codificador que traduce el número decimal de la tecla oprimida a un código binario tal como el BCD – 8421. La CPU ejecuta la operación solicitada en binario y entrega el resultado en binario. El decodificador traduce del código binario de la CPU a un código especial que iluminará los segmentos que correspondan en la pantalla display entregando el resultado en sistema decimal.

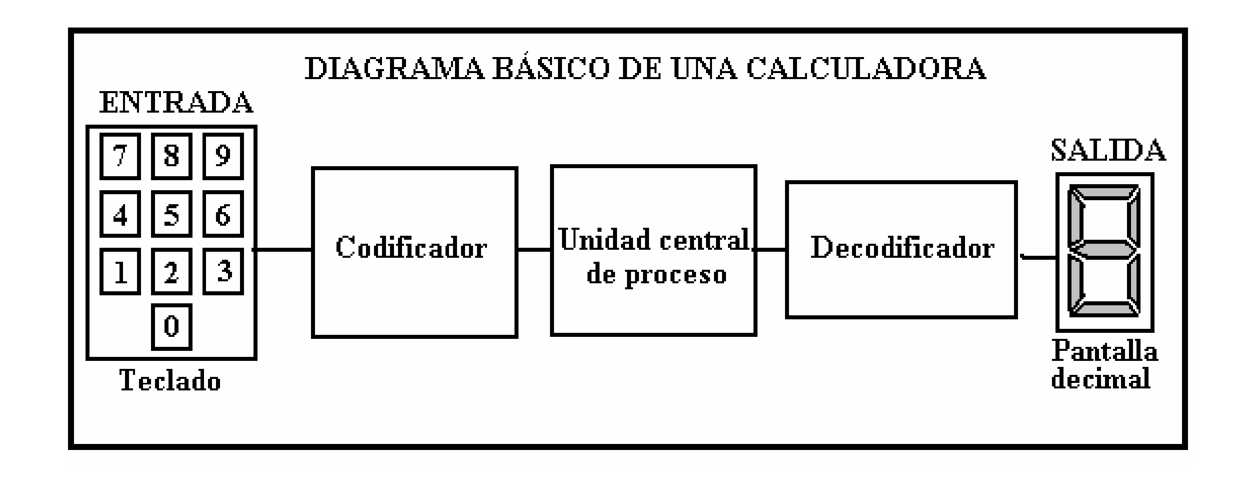


**FIG. A** **Codificador comercial de prioridad Decimal a BCD**

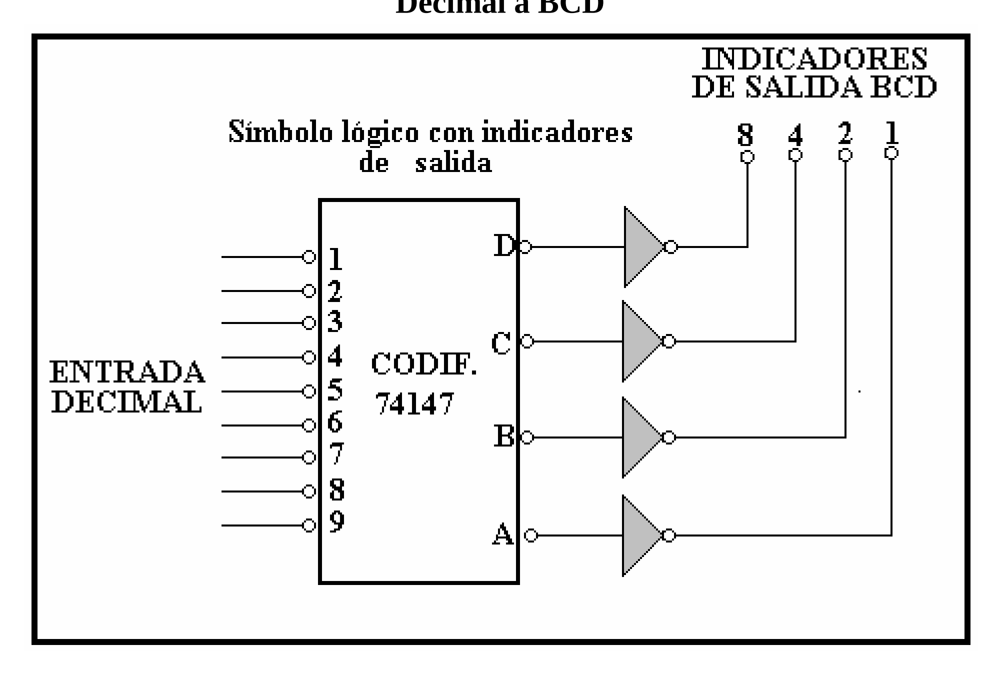



**FIG. B** **TABLA DE LA VERDAD**

**SÍMBOLO LÓGICO CON INDICADORES DE SALIDA**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | D | C | B | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | A | A | A | A | A | A | A | A | A | A | A |
| X | X | X | X | X | X | X | X | B | B | A | A | B |
| X | X | X | X | X | X | X | B | A | B | A | A | A |
| X | X | X | X | X | X | B | A | A | A | B | B | B |
| X | X | X | X | X | B | A | A | A | A | B | B | A |
| X | X | X | X | B | A | A | A | A | A | B | A | B |
| X | X | X | B | A | A | A | A | A | A | B | A | A |
| X | X | B | A | A | A | A | A | A | A | A | B | B |
| X | B | A | A | A | A | A | A | A | A | A | B | A |
| B | A | A | A | A | A | A | A | A | A | A | A | B |

**A = NIVEL LOGICO ALTO**
**B = NIVEL LÓGICO BAJO**
**X = IRRELEVANTE**

## CODIFICADORES

Tal como se indicó, la función del codificador en una calculadora es la de traducir una entrada decimal a código binario BCD (8421).

La figura A muestra un codificador de 10 entradas y 4 salidas, el cual puede tener solo una entrada activa, la que a su vez produce una salida única.

Los círculos de las entradas significan que estas podrán ser activadas por el 0 lógico o bajo; en cambio, los círculos de las salidas significan que en estado inactivo, las salidas permanecen en 1 lógico o alto y cuando son activadas pasan a 0 lógico. Los inversores de la salida permiten invertir la salida a su forma normal.

Otra características de este codificador es que no posee entrada cero. La entrada decimal cero significa que la salida D C B A será 1111 y por la presencia de los inversores, la salida final aplicada los indicadores BCD será 0 0 0 0. Cuando esto sucede, se dice que las entradas están flotando. En este caso flotan en alto ó 1 lógico.

El codificador de la Fig. A es un dispositivo TTL de serie la 74147 cuya tabla de verdad es la que se observa en la Fig. B.

La primera línea de la tabla de verdad es para cuando no hay entradas. Cuando todas las entradas flotan en alto, las salidas también flotan en alto, lo que como se indicó anteriormente se interpreta como 0 0 0 0 en los indicadores BCD.

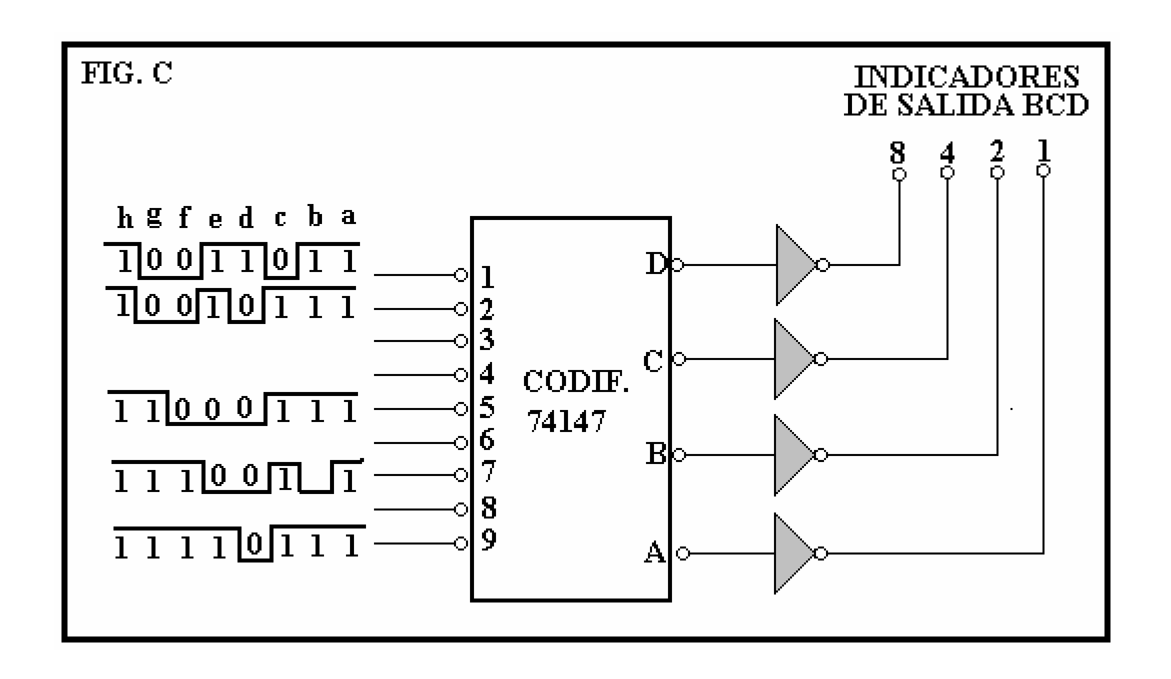
La segunda línea muestra la entrada decimal 9 activada por un 0 lógico, lo que produce BAAB en las salidas DCBA; los cuatros inversores invierten BAAD, obteniéndose de ellos la combinación lógica ABBA, que al ser aplicada a los indicadores BCD, éstos interpretarán como 1001 que es la forma de representar al número 9 decimal en BCD.

La segunda línea de la tabla de verdad también muestra las entradas 1 al 8 marcadas con una X que significa irrelevante. Una entrada irrelevante puede ser alto ó bien bajo. Este codificador tiene un mecanismo de prioridad que activa el número mayor que tiene entrada baja. Si por cualquier motivo en el momento de activar el 9 se activara simultáneamente cualquier número menor, por ejemplo el 5, la salida final sería 1001, ya que el codificador solo activa la salida del número mayor. Si en cambio, se activa el número 5, según la tabla de verdad los números superiores a éste quedan en estado lógico alto (1 lógico) y no en estado irrelevante.

Esta característica permite mantener para todos los números, la prioridad que activa al número mayor que tiene entrada bajo (0 lógico).

**PROBLEMA:** Haga una lista de las salidas en los indicadores BCD – 8421 para cada una de las ocho combinaciones de entrada que se muestran en la Fig. C.

Se debe recordar que el mecanismo de prioridad activa al número mayor que tenga entrada baja o 0 lógico.




**SOLUCION:**

| | | | |
|---|---|---|---|
| PULSO a = | PULSO c = | PULSO e = | PULSO g = |
| PULSO b = | PULSO d = | PULSO f = | PULSO h = |

## DECODIFICADORES: BCD A DECIMAL

Podría decirse que un decodificador es lo opuesto a un codificador. Si se invierte el proceso descrito para los codificadores se produciría un decodificador que traduce del código BCD a decimales.

La Fig. D muestra un decodificador comercial TTL de serie 7442. Las cuatro entradas BCD a la izquierda del símbolo lógico se representan como D, C, B, A, en donde la entrada D corresponde a los 8, mientras que las restantes entradas C, B y A corresponden a los 4, 2 y 1 respectivamente. El nivel 1 lógico o alto activa las entradas. A la derecha se encuentran las diez líneas de salida y los círculos adjuntos al símbolo lógico indican que las salidas del decodificador flotan en alto, cambiándose a nivel bajo ó 0 lógico al ser activadas.

El agregado de los inversores, permite que se inviertan los niveles lógicos de las salidas del decodificador, con el objeto de manejar las luces de los indicadores de salida. Por lo tanto una salida activa (0 lógico) en el decodificador, aparecerá como un 1 lógico en los indicadores de salida, permitiendo que se encienda la luz respectiva.

La interpretación del encendido de las luces en los indicadores de salida se debe hacer de acuerdo al código BCD – 8421. Por ejemplo, si se encienden las dos luces de la derecha corresponderá al Nº 3 decimal.

**FIG. D Símbolo lógico de un decodificador comercial**

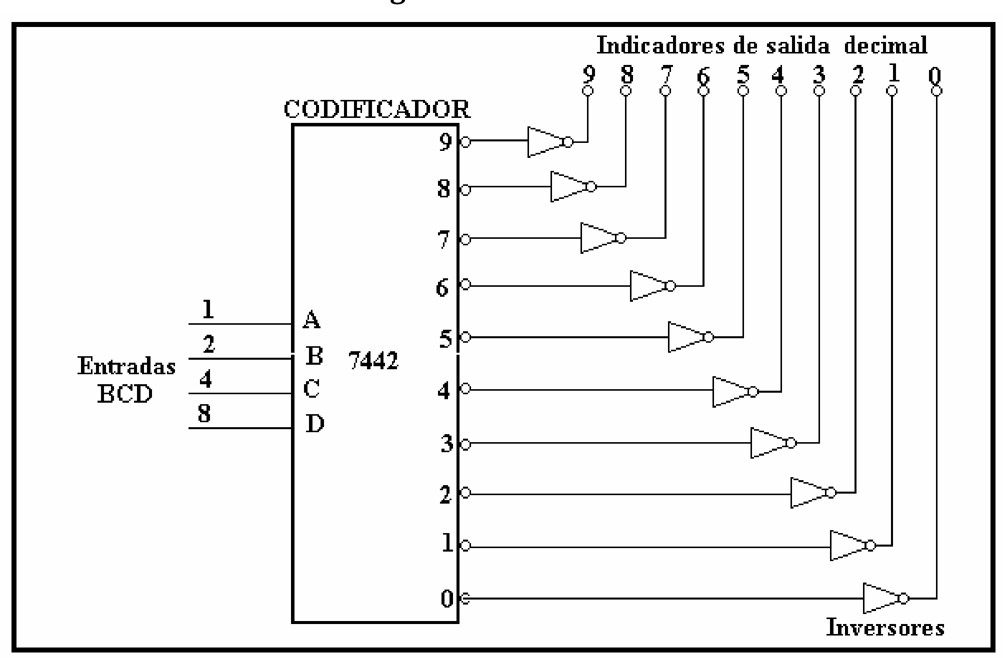



**FIG. E** **<u>TABLA DE VERDAD DEL DECODIFICADOR 7442</u>**

| Línea | Nº | Entradas BCD | | | | Salida Decimal | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **D** | **C** | **B** | **A** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| Línea 1 | 0 | B | B | B | B | B | A | A | A | A | A | A | A | A | A |
| Línea 2 | 1 | B | B | B | A | A | B | A | A | A | A | A | A | A | A |
| Línea 3 | 2 | B | B | A | B | A | A | B | A | A | A | A | A | A | A |
| Línea 4 | 3 | B | B | A | A | A | A | A | B | A | A | A | A | A | A |
| Línea 5 | 4 | B | A | B | B | A | A | A | A | B | A | A | A | A | A |
| Línea 6 | 5 | B | A | B | A | A | A | A | A | A | B | A | A | A | A |
| Línea 7 | 6 | B | A | A | B | A | A | A | A | A | A | B | A | A | A |
| Línea 8 | 7 | B | A | A | A | A | A | A | A | A | A | A | B | A | A |
| Línea 9 | 8 | A | B | B | B | A | A | A | A | A | A | A | A | B | A |
| Línea 10 | 9 | A | B | B | A | A | A | A | A | A | A | A | A | A | B |
| Línea 11 | NO | A | B | A | B | A | A | A | A | A | A | A | A | A | A |
| Línea 12 | NO | A | B | A | A | A | A | A | A | A | A | A | A | A | A |
| Línea 13 | NO | A | A | B | B | A | A | A | A | A | A | A | A | A | A |
| Línea 14 | NO | A | A | B | A | A | A | A | A | A | A | A | A | A | A |
| Línea 15 | NO | A | A | A | B | A | A | A | A | A | A | A | A | A | A |
| Línea 16 | NO | A | A | A | A | A | A | A | A | A | A | A | A | A | A |

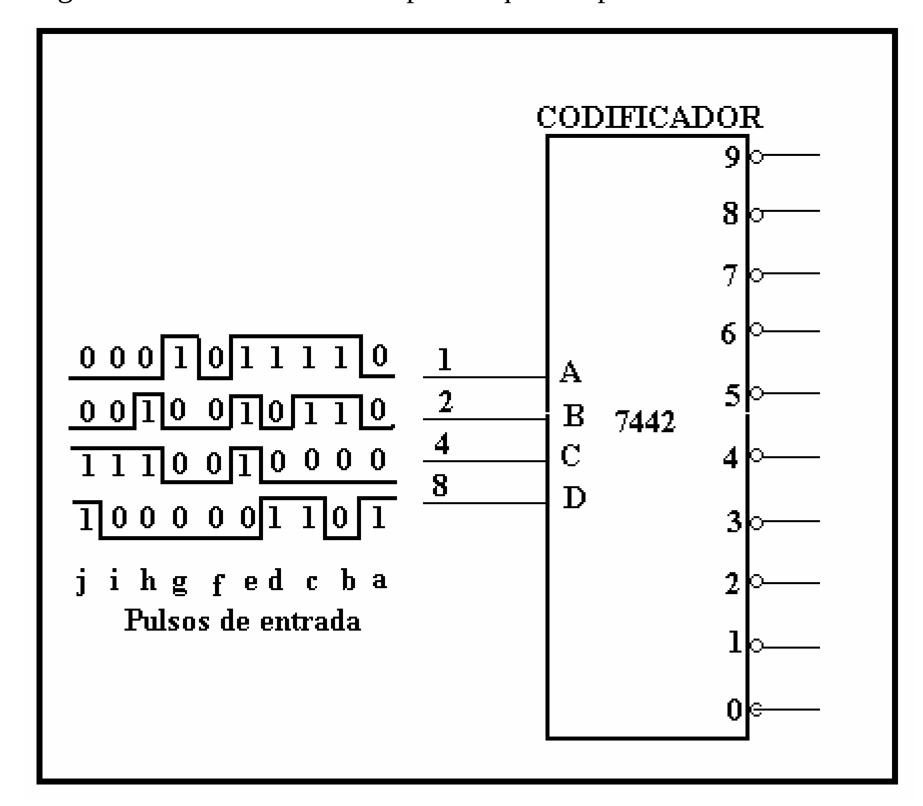
**A = ALTO**
**B = BAJO**

La tabla de verdad para el decodificador 7442 se muestra en la Fig. E. La primera línea, que representa al 0 decimal, muestra a todas las entradas en Bajo (B). Con la entrada BBB (0000), la salida decimal 0 del decodificador se activa a un estado lógico B. El inversor del extremo inferior convierte esta salida a un estado lógico A, lo que permite lograr el encendido del indicador de la salida decimal correspondiente al Nº 0. Los demás indicadores permanecen apagados.

De la misma manera, la quinta línea que representa al decimal Nº4, muestra la entrada BCD como BABB (0100). Esto hace que la salida decimal 4 del decodificador se active a un estado lógico B, que se invierte iluminando al indicador decimal Nº4. De esto se deduce que este decodificador tiene entradas activadas en alto (A y salidas activadas en Bajo (B).

Considérese la línea 11 de la Fig. E, cuya entrada es ABAB 1010 y que representa al decimal. Ya que el código de salida no contiene este número, se considera que esta es una entrada no válida y ningún indicador de salida podrá iluminarse. Se puede notar en la tabla de verdad que las 6 últimas líneas de la tabla de verdad muestran entradas no válidas, por lo tanto con cualquiera de estas entradas aplicadas al decodificador los indicadores permanecerán apagados.

**PROBLEMA:** Indique cuales serán las salidas activas del decodificador 7442 para cada una de las siguientes combinaciones de pulsos que se aplican en sus entradas A -B-C-D.



**SOLUCION:**

PULSO a = PULSO c = PULSO e = PULSO g =
PULSO b = PULSO d = PULSO f = PULSO h =

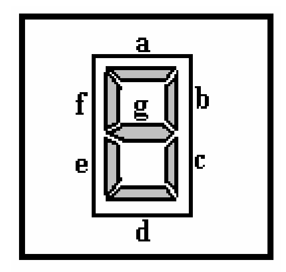
PULSO i =
PULSO J =

## DECODIFICADOR BCD A CODIGO DE SIETE SEGMENTOS

Un trabajo común en un circuito digital, es traducir lenguaje de máquina a números decimales y una forma usual de hacerlo es mostrar los números decimales en una pantalla de siete segmentos, las cuales pueden construirse de varias maneras. Como por ejemplo se pueden señalar los siguientes tipos:

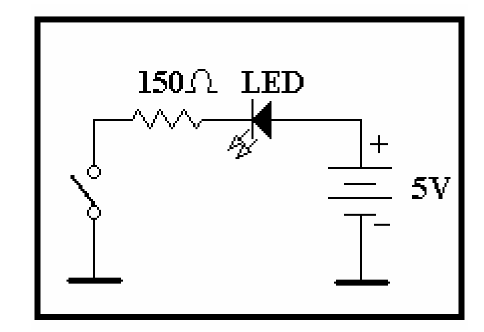
a) Pantalla de filamento incandescente, que es similar a una lámpara común.
b) Pantalla de emisión de gas, que opera con altos voltajes y da un brillo anaranjado.
c) Pantalla fluorescente, que opera con bajos voltajes y da un brillo verdoso.
d) Pantalla LED con diodos emisores de luz. Esta es una de las más comunes y emite un brillo rojizo característico.
e) Pantalla de cristal líquido (LCD), que opera con bajos voltajes y produce los números en un color gris, plateado o negro.

### IDENTIFICACIÓN DE SEGMENTOS




En la Figura se puede observar la identificación que se asigna a los segmentos de la pantalla decimal.

### FUNCIONAMIENTO DE UN DIDODO EMISOR DE LUZ



La Figura señala una batería de 5 volt conectada a un LED simple. Cuando se cierra el switch una corriente cercana a los 20 mA flu yen por el circuito iluminando al diodo LED.

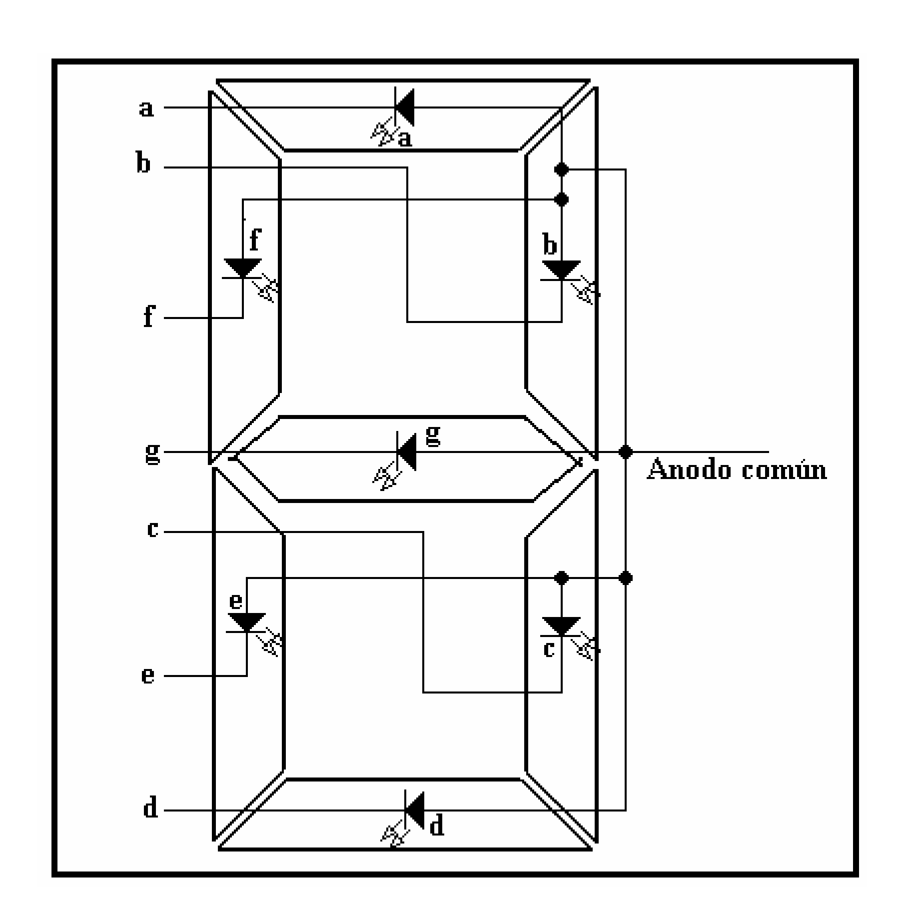
El resistor de 150 ohms se coloca en el circuito para limitar la corriente circulante a 20 mA.

Debido a que los diodos LED admiten una sola polaridad, el catodo debe quedar conectado hacia el negativo (GND) d e la batería y anodo hacia el positivo.

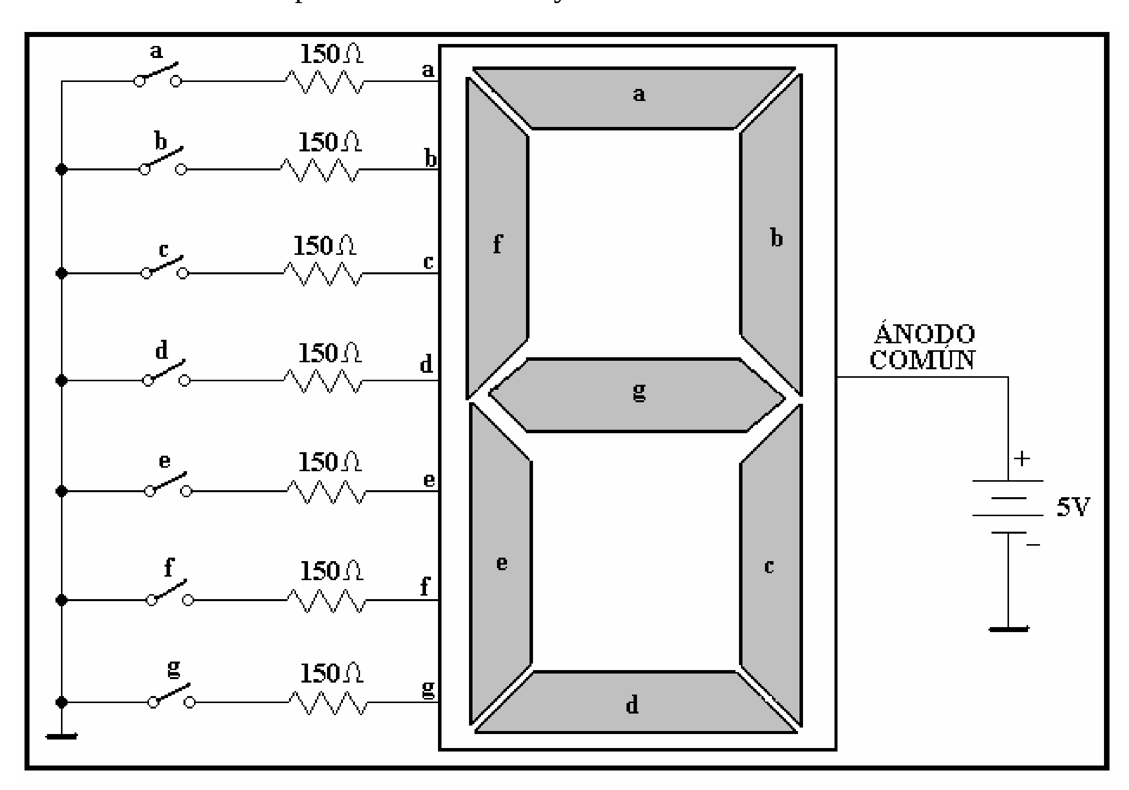
Normalmente la polarización directa que acepta un diodo LED fluctúa entre 1,7 V y 2,2 V según el tipo.

La Figura siguiente muestra una pantalla LED de siete segmentos. Cada segmento (de a hasta g) contiene un diodo LED, los cuales a su vez se encuentran con sus anodos unidos, por lo que este dispositivo recibe el nombre de pantalla o display tipo LED de ánodo común.

Las entradas de la izquierda corresponden a cada uno de los cátodos LED, las cuales podrán ser conectadas a la salida de un decodificador BCD a decimal que active los segmentos que permitan representar el número procesado.



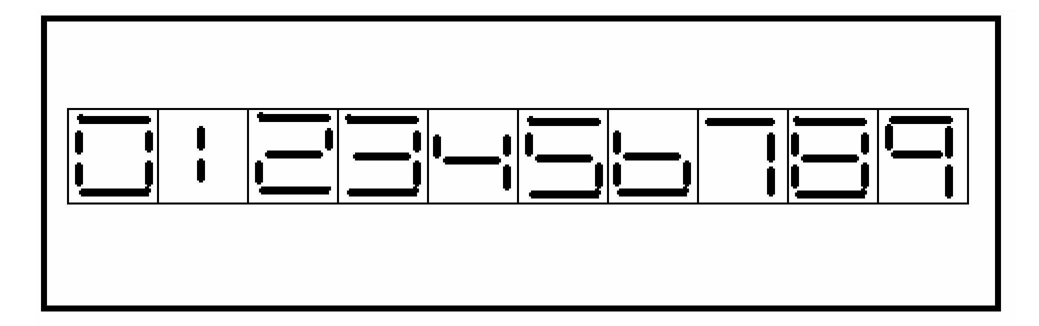
La Fig siguiente muestra la forma como se puede conseguir el encendido de los siete segmentos que forman la pantalla. Se puede deducir que de la combinación de encendido de los switch de entrada, se podrá lograr que la pantalla represente cualquiera de los números decimales comprendidos entre el 0 y el 9 inclusive.




**PROBLEMA**: Según la figura anterior, señale cuales switch de entrada deberán cerrarse para dar forma a los siguientes números:

| | |
|---|---|
| **1 =** | **5 =** |
| **2 =** | **6 =** |
| **3 =** | **7 =** |
| **4 =** | **8=** |

**REPRESENTACIÓN DE LOS NUMEROS DECIMALES EN LA PANTALLA**

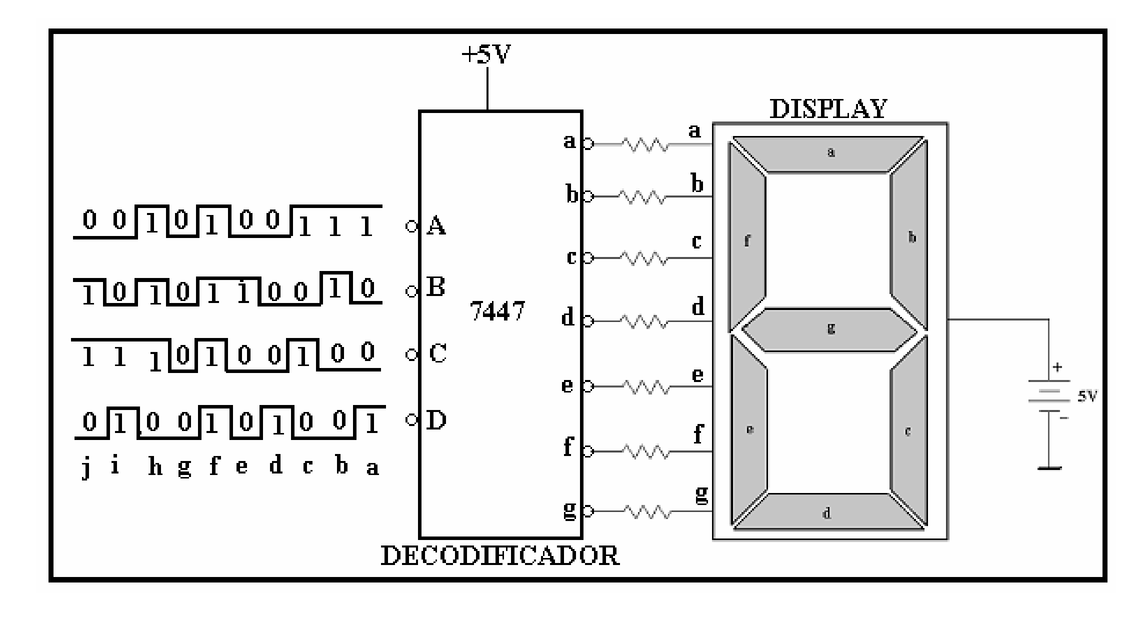


## TABLA DE VERDAD SIMPLIFICADA DEL DECODIFICADOR 7447 – BCD PARA SIETE SEGMENTOS

| **Decimal** | **Entradas** | **Salidas** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **D C B A** | **A** | **b** | **c** | **d** | **e** | **f** | **g** |
| 0 | B B B B | ON | ON | ON | ON | ON | ON | OFF |
| 1 | B B B A | OFF | ON | ON | OFF | OFF | OFF | OFF |
| 2 | B B A B | ON | ON | OFF | ON | ON | OFF | ON |
| 3 | B B A A | ON | ON | ON | ON | OFF | OFF | ON |
| 4 | B A B B | OFF | ON | ON | OFF | OFF | ON | ON |
| 5 | B A B A | ON | OFF | ON | ON | OFF | ON | ON |
| 6 | B A A B | OFF | OFF | ON | ON | ON | ON | ON |
| 7 | B A A A | ON | ON | ON | OFF | OFF | OFF | OFF |
| 8 | A B B B | ON | ON | ON | ON | ON | ON | ON |
| 9 | A B B A | ON | ON | ON | OFF | OFF | ON | ON |
| 10 | A B A B | OFF | OFF | OFF | ON | ON | OFF | ON |
| 11 | A B A A | OFF | OFF | ON | ON | OFF | OFF | ON |
| 12 | A A B B | OFF | ON | OFF | OFF | OFF | ON | ON |
| 13 | A A B A | ON | OFF | OFF | ON | OFF | ON | ON |
| 14 | A A A B | OFF | OFF | OFF | ON | ON | ON | ON |
| 15 | A A A A | OFF | OFF | OFF | OFF | OFF | OFF | OFF |

**B = BAJO** **A = ALTO**
**OFF = APAGADO** **ON = ENCENDIDO**

**PROBLEMA**: Señale cuales son los Nº o indicaciones que representará la pantalla decimal para las siguientes combinaciones binarias en la entrada del decodificador 7447 -BCD para siete segmentos.



**SOLUCION:**

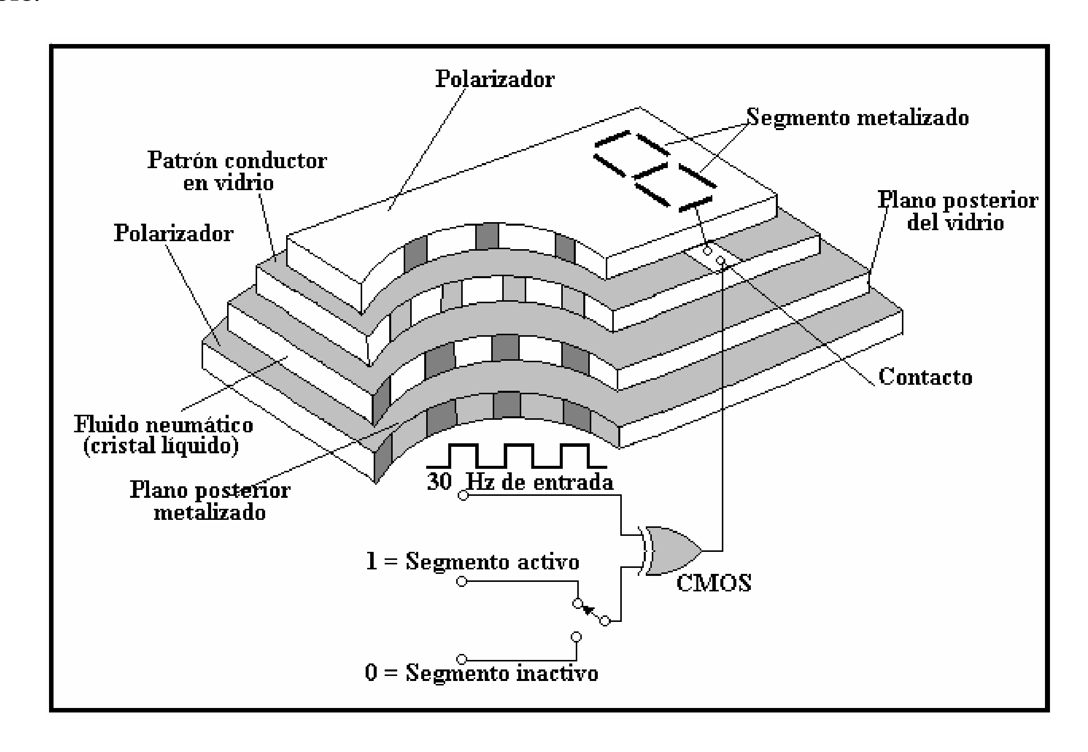
PULSO a = PULSO f =
PULSO b = PULSO g =
PULSO c = PULSO h =
PULSO d = PULSO i =
PULSO e = PULSO j =

## PANTALLA DIGITAL DE CRISTAL LIQUIDO LCD

El visor de cristal líquido opera en muy diferente formas que el tipo LED. Un LED genera luz de salida en tanto que un LCD controla la luz. El LCD necesita una entrada de luz para ser visto mientras que el LED produce su propia luz. La principal ventaja del anunciador de cristal líquido es que consume una cantidad extremadamente pequeña de potencia.

En la figura se muestra el diagrama de un LCD común: La construcción es de un LDC de efecto de campo. Cuando se energiza el segmento LCD aparece negro en comparación con el resto de la superficie brillante. El segmento "e" está energizado en la ilustración. En resto de los segmentos son casi invisibles.

La clave de operación del LCD es el fluido nemático, el cual es un cristal líquido. Esta intercalado entre dos placas de vidrios. Se aplica voltaje a través del fluido nemático de la parte superior metalizada del segmento al plano metalizado posterior. Entonces el fluido transmite la luz en forma diferente y el segmento energizado del visor se vuelve visible.



Los anunciadores de cristal líquido se energizan con voltaje de CA; en la figura se observan las entradas.
En la figura se aprecia la construcción y operación de un LCD de efecto de campo con siete segmentos.

## RESUMEN DE FAMILIAS LOGICAS

## FAMILIAS BIPOLARES

### 1) TTL (TRANSISTOR LOGIC).-

En la más amplia familia de circuitos integrados digitales. Su encapsulamiento es de bajo costo y es la de mayor difusión en el mercado.

**PROPIEDADES:**

- Funciones lógicas por medio de transistores multi – emisor.
- Alta velocidad de operación: Tiempo de propagación típico (10 ns / gate)
- Baja inmunidad al ruido.
- Consumo elevado (10 mW / gate)

**TIPOS ESPECIALES**

SERIE L: Para bajo consumo (1 mW / gate).
SERIE S: Para altas velocidades de operación (80 MHz). Tecnología Schottky.
SERIE LS: Para bajo consumo y altas velocidades (25 MHz – 2 mW / gate)

## TECNOLOGÍA SCHOTTKY

La Tecnología Schottky se refiere a un tipo de diodos empleados en los CI – TTL denominados diodos de efecto Schottky de barrera (SBD), formados por una juntura metal – semiconductor. El comportamiento eléctrico de un SBD es similar al de un diodo convencional, pero se distingue de este último por operar con cargas mayoristas y por su menor voltaje Vd en polarización directa, el cual es del orden de 0,3 volt.

El empleo de este tipo de diodos permite reducir considerablemente los tiempos de propagación de BIT en los CI – TTL, lo que hace que estos últimos puedan lograr altas velocidades de operación.

**2) ECL (EMITTER CUPLED LOGIC). -**

Familia de integrados digitales para altísimas velocidades de operación. (también conocida como CML, Lógica- Modo-Corriente).

**PROPIEDADES:**

- Las funciones lógicas son obtenidas a través de transistores en paralelo con los emisores acoplados.
- Tiempos de propagación de 1 ns / gate, obtenidos con transistores operando en la región no saturada.
- Baja inmunidad al ruido
- Consumo elevado (30 mW / gate).

**3) L SL (LOW SPEED LOGIC). -**

Familia de integrados con alta inmunidad al ruido, para aplicaciones en control industrial o en ctos sujetos a altos niveles de ruido.

**PROPIEDADES:**

- Trabaja con voltajes más altos que la familia TTL (15 V), soporta niveles de ruido 4 volt.
- La inmunidad a ruidos estáticos o dinámicos se debe a un diodo zener colocado internamente.
- En aplicaciones más críticas, la inmunidad al ruido dinámico puede ser reforzada por un capacitor colocado externamente.

**4) HTL (HIGH THRESHOLD LOGIC). -**

Familia con tecnología para aplicaciones en sistemas donde se requiere alta inmunidad al ruido (como la familia LSL), pero sin la posibilidad de colocación de un capacitor externo.

**5) RTL (RESISTOR TRANSISTOR LOGIC)**

**6) DTL (DIODE TRANSISTOR LOGIC)**

## FAMILIAS DE MOS

### 1) C – MOS (COMPLEMENTARY METAL OXIDE SEMICONDUCTOR)

Familia de baja disipación de potencia (cercana al uW), con voltajes de alimentación no críticos (3 a 18 v), además de alta inmunidad al ruido.

**PROPIEDADES:**

- Los circuitos integrados C-MOS trabajan en régimen complementario, es decir, dos transistores en serie, de manera que siempre uno de ellos no este al corte.
- Esta configuración posibilita un consumo mínimo (0, 01 mW / gate a 1 KHz)
- Alta velocidad de operación (20 ns / gate)
- Compatible con TTL

2) P – MOS
3) N – MOS

Las familias bipolares se consiguen con CI que contienen partes comparables a los transistores bipolares discretos, diodos y resistores.

Otro grupo de familias de CI digitales utiliza la tecnología de semiconductor de metal – oxido (MOS) que contiene partes comparables a transistores de efecto de campo de compuerta aislada (IGFETS).

Los fabricantes comúnmente clasifican a los circuitos integrados en grupos basados en la complejidad de los circuitos:

**a) L S I (INTEGRACION A GRAN ESCALA). -**

Cien o más compuertas equivalentes o circuitos de complejidad similar.

**b) MSI (INTEGRACIÓN A MEDIANA ESCALA)**

Doce o más compuertas equivalentes o circuitos de complejidad similar.

**c) S S I (INTEGRACIÓN A PEQUEÑA ESCALA)**

Corresponde a CI de menor complejidad que los MSI.

**d) VLSI (INTEGRACIÓN A MUY GRANDE ESCALA)**

Mil o más compuertas o circuitos de complejidad similar.

## TIPOS DE LOGICA

Se considera lógica positiva cuando en un sistema digital el estado lógico **“1”** corresponde a un voltaje alto mayor que cierto valor VH mínimo y el estado lógico **“0”** corresponde a un voltaje bajo menor que VL máximo; en cambio, se considera lógica negativa cuando el estado lógico **“0”** corresponde al nivel alto del voltaje y el estado lógico **“L”** al nivel bajo del voltaje.

De los dos tipos de lógica utilizados, el más común es el de **LOGICA POSITIVA**.

| Lógica positiva | | Lógica negativa | |
|---|---|---|---|
| V | | V | |
| Vh mínimo | "1" lógico | Vh mínimo | "0" lógico |
| Vl máximo | "0" lógico | Vl máximo | "1" lógico |

# *PRÁCTICA*

# ELECTRÓNICA

# INDUSTRIAL

## ENVÍO 10



# ELECTRÓNICA DIGITAL

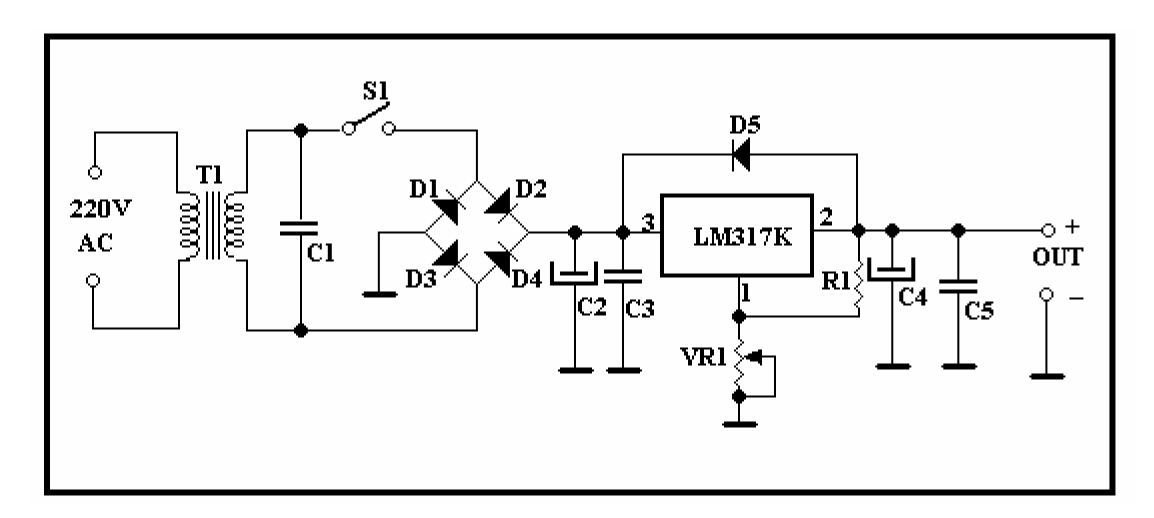
**Consideraciones iniciales:**

Para lograr un adecuado trabajo práctico en circuitos digitales, como por ejemplo el armado de los mismos, características de los circuitos integrados utilizados, alimentación del circuito y comprobación del funcionamiento de acuerdo a tablas de verdad, se hacen necesarios los siguientes materiales:

**1.-fuente de poder regulada de 0 a 37Vcc.**
**2.-protoboard.**
**3.-punta de prueba lógica.**
**4.-tester digital.**
**5.-manual de caracteríscas.**

1.-**FUENTE DE PODER REGULADA DE 1,2 A 37VCC.**

A través de una fuente regulada, como la que se muestra a continuación, se pueden obtener los distintos voltajes que se requieren para la polarización de los circuitos integrados de las distintas lógicas.

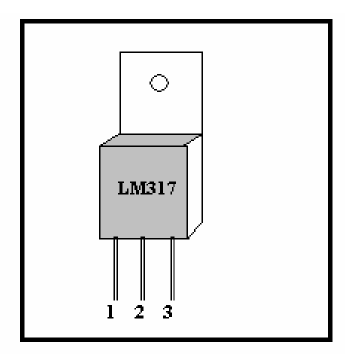



**Materiales necesarios para el armado.**

| | |
|---|---|
| T1 | :Transformador de 30V / 2A. |
| D1/D5 | :4 Diodos 1N4007. |
| C1/C3/C5 | :3 condensadores de 0,33 UF x 50V. |
| C2 | :Condensador de 3.300 UF x 63V. |
| C4 | :Condensador de 2,2 UF x 50V. |
| VR1 | :Potenciómetro de 10K con interruptor (eje corto y plástico). |
| R1 | :Resistencia de 270 /1/2W |
| IC | :Circuito integrado LM 317. |

Es importante destacar que a este diseño se le pueden agregar los accesorios que el interesado estime convenientes (Voltímetro, amperíme tro, diodos leds, etc.)

**Identificación de los terminales del CI LM317.**



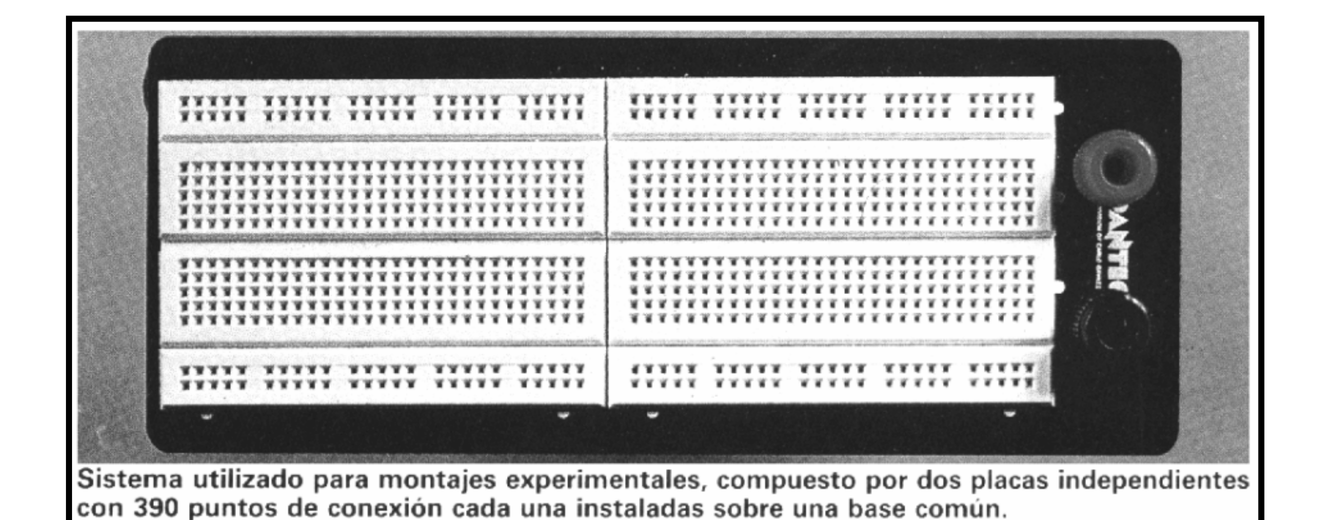

**1 = Adj. (Potenc.)**
**2 = Out.**
**3 = In. (40Vcc máx.)**

**Precauciones en el uso.**

Antes de encender la fuente, verificar que los terminales de salida de la misma no se encuentran cortocircuitados, puesto de ser así, se quema la fuente.

**2.-PROTOBOARD.-**

Un protoboard de experimentación consiste en una placa de material plástico que contiene un gran número de orificios de inserción, sobre los que se encuentran, por la cara inferior, unos puntos de contacto metálicos que trabajan por el sistema de presión sobre el terminal que se aplique a ellos.



Sistema utilizado para montajes experimentales, compuesto por dos placas independientes con 390 puntos de conexión cada una instaladas sobre una base común.

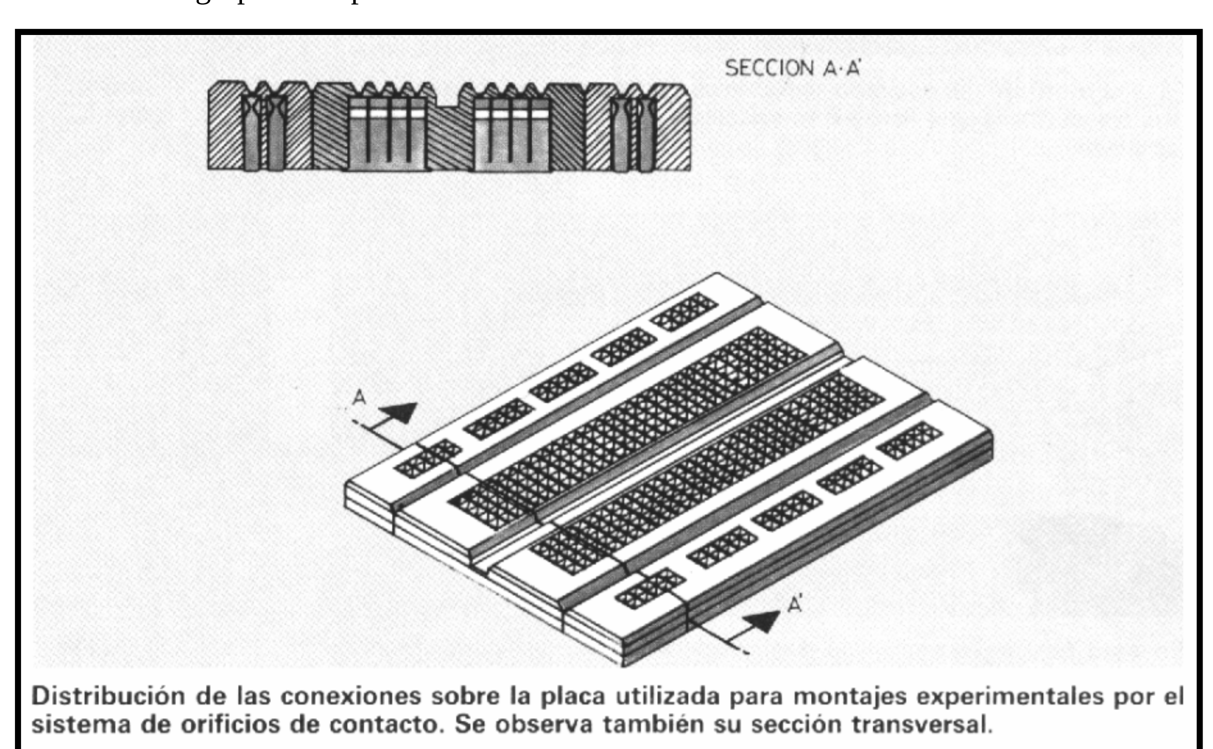
La gran ventaja del protoboard radica en que su uso permite realizar el montaje de diversos diseños electrónicos o experimentar los propio s, sin tener que recurrir al cautín y soldadura para efectuar la conexión de componentes.

**Características del protoboard K-H modelo GL-23**

a) Posee un total de 1.580 puntos de contacto.

b) Dispone de tres columnas delgadas de 100 puntos cada una. Cada co lumna tiene interconexión en grupos de 25 puntos verticales.

c) Tiene 4 columnas anchas de 320 puntos cada una. Cada columna posee interconexión horizontal en grupos de 5 puntos.



Distribución de las conexiones sobre la placa utilizada para montajes experimentales por el sistema de orificios de contacto. Se observa también su sección transversal.

## 3.-PUNTA DE PRUEBA LÓGICA.

Hoy en día es muy común que en los aparatos electrónicos encontremos circuitos integrados digitales. En las mediciones que realicemos en ellos solo nos interesa conocer los niveles lógicos presentes en las salidas; es decir, saber si estamos ante la presencia de un nivel lógico alto (1), bajo (0) o frente a un tren de pulsos.
Para poder efectuar este tipo de mediciones, lo más lógico y útil es emplear una punta de prueba lógica. Esta punta de prueba lógica es muy fácil de construir y de bajo costo.

En nuestra práctica nos será de mucha utilidad y su metodología de operación se explica de la siguiente manera:

a) Cuando la punta está en el aire o conectada en un punto muerto del circuito bajo prueba, D1 estará apagado y D2 también estará apagado

b) Si la punta de prueba está en alto, D1 estará encendi do y D2 continuará apagado.

c) Si la punta de prueba está en bajo, D1 estará apagado y D2 encendido (por ser un diodo bicolor R – V ).

d) Si en la punta está presente un tren de pulsos de baja frecuencia, D1 y D2 estarán encendidos de forma intermitente.

e) Si se presenta en la punta de prueba un tren de pulsos de alta frecuencia, D1 y D2 estarán encendidos, adquiriendo un color anaranjado.

Este circuito puede ser montado dentro de un lápiz en desuso, otorgándole así mayor movilidad.
Con lo anteriormente expuesto podemos resumir la operación del circuito de la siguiente forma:

**-Led verde encendido--------------------“0”**

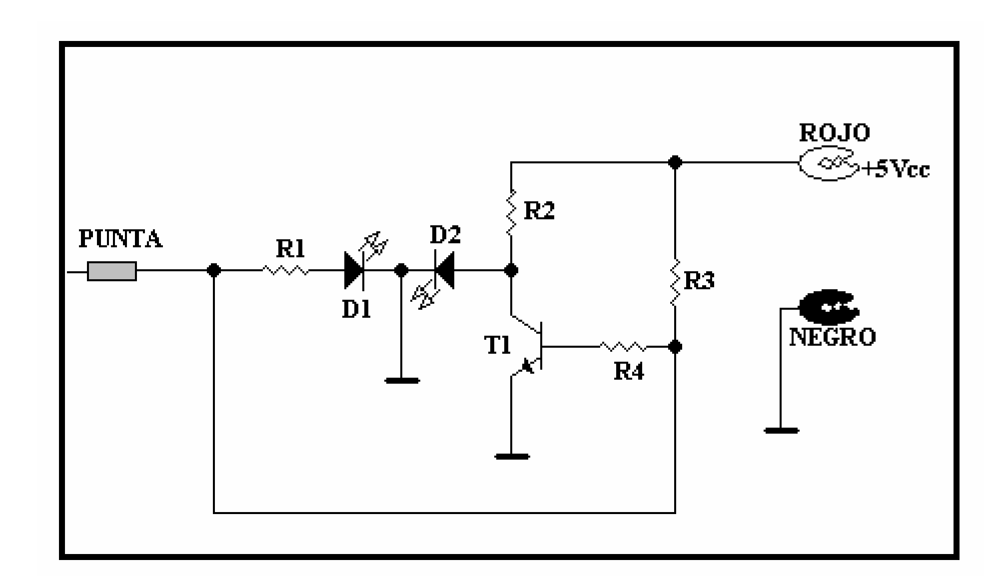
**-Led rojo encendido----------------------“1”**

**-Leds apagados----------------------------“no conectada”**

**-Leds intermitentes ------------------------“Tren de pulsos de baja frecuencia”**

**-Leds anaranjado ------------------------“Tren de pulsos de alta frecuencia”**

## Circuito eléctrico de la punta de prueba.




## Armado de la punta de prueba lógica.

a) De acuerdo al circuito eléctrico, confeccionar el pictograma.

b) Transferir el pictograma a una placa de circuito impreso.

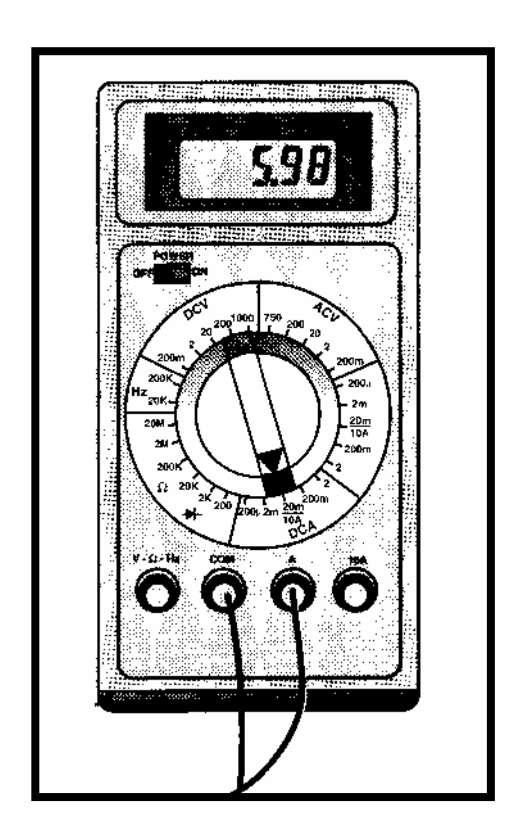
c) Montar los elementos de acuerdo al circuito.

## LISTA DE MATERIALES.

| | |
|---|---|
| R1 | :Resistencia de 220 ohms-1/4 W. |
| R2 | :Resistencia de 220 ohms-1/4 W. |
| R3 | :Resistencia de 1,2 K. ohms-1/4 W. |
| R4 | :Resistencia de 100 ohms-1/4 W. |
| T1 | :Transistor BC548. |
| D1/D2 | :Leds de dos colores (V/R), con cátodo común. |

## 4.-TESTER DIGITAL.

Permite verificar que el voltaje de alimentación que polariza a los CI es acorde a su fam ilia lógica. Se caracteriza en la gran precisión que se obtiene de la medición.




## 5.-MANUAL DE CARACTERÍSTICAS.

Permite averiguar las principales características del circuito integrado y  su reemplazo en el caso de avería.

## COMPROBACIÓN DE COMPUERTAS LÓGICAS E INTEGRADOS TTL.

**1.-Verificar tabla de la verdad del CI 7404, que actúa como séxtuple compuerta inversora (Hex inverter).**

| A | Y |
|---|---|
| 0V | Vcc |
| Vcc | 0V |

**Para efectuar esta prueba, realice el siguiente con exionado:**

a)Por tratarse de un integrado de la familia TTL, regule la fuente a 5Vcc.

b)Alimente el pin 14 del integrado con +5Vcc y el pin 7 con el negativo de la fuente.

c)El positivo de la fuente será considerado como "1 lógico" y el negativo como "0 lógico" .

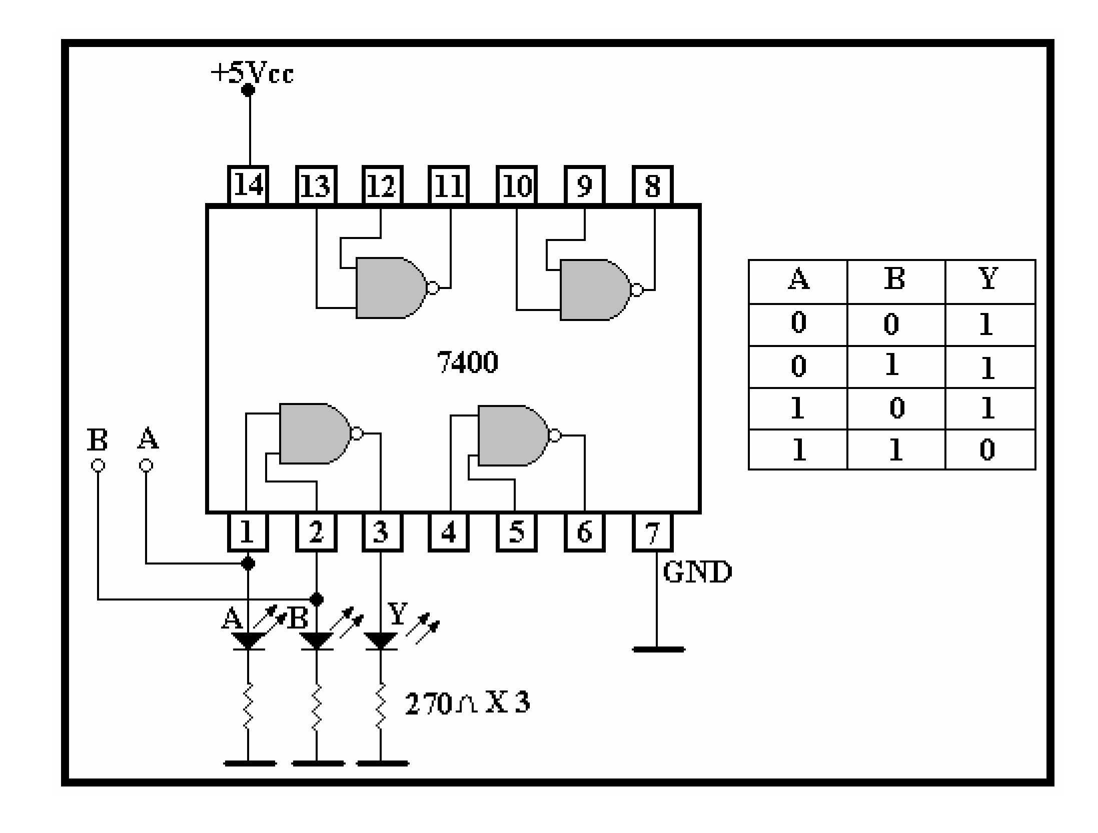
d)Conecte la entrada de la primera compuerta a "1 lógico" ya través de la punta de prueba lógica verifique el nivel de la salida. Anote este nivel en la tabla de la verdad.

e)Conecte la entrada de la primera compuerta a "0 lógico" y a través d e la punta de prueba lógica verifique el nivel de la salida. Anote este nivel en la tabla de la verdad.

f)Repita estas pruebas con cada una de las restantes compuertas inventoras.

**Conclusiones:**..................................................................................................................
.......................................................................................................................................
.......................................................................................................................................

**2.-Verificar tabla de la verdad del CI 7400, que actúa como cuadruple compuerta NAND de dos entradas cada una.**




| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

a) Inserte el CI 7400 entre dos columnas anchas del protoboard.

b) Regule la fuente a 5Vcc.

c) Alimente el pin 14 del integrado con +5Vcc y el pin 7 con el negativo de la fuente

d) Conecte la entrada A de la primera compuerta "0 lógico" y la entrada B a "0 lógico". Luego, a través de la punta de prueba lógica verifique el nivel de salida y anótelo en la tabla de la verdad.

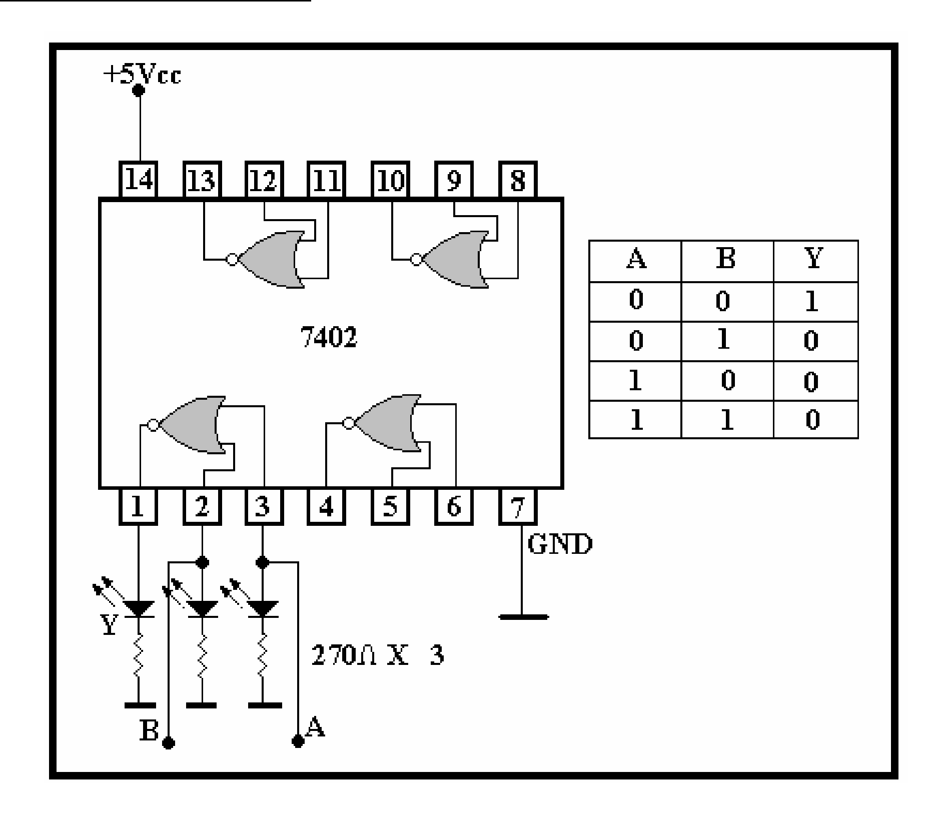
e) Conecte la entrada A de la primera compuerta a "1 lógico" y la entrada B a "0 lógico". Luego, verifique el nivel de salida y anótelo en la tabla de la verdad.

f) Conecte la entrada A de la primera compuerta a "0 lógico" y la entrada B a "1 lógico". Luego, verifique el nivel de salida y anótelo en la tabla de la verdad.

g) Conecte ambas entradas (A y B) a "1 lógico" y luego, a través de la punta de prueba lógica verifique el nivel de salida y anótelo en la tabla de la verdad.

h) Repita estas pruebas en cada una de las restantes compuertas que conforman al integrado 7400.

**3.-<u>Verificar tabla de la verdad del CI 7402, que actúa como cuadruple compuerta NOR de dos entradas cada una.</u>**




| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

a) Inserte el CI 7402 entre dos columnas anchas del protoboard.

b) Regule la fuente a 5Vcc.

c) Alimente el pin 14 del integrado con +5cc y el pin 7 con el negativo de la fuente.

d) Conecte la entrada A de la primera fuente a “0 lógico” y lo mismo haga con la entrada B verifique posteriormente el nivel de entrada y anótelo en la tabla de la verdad.

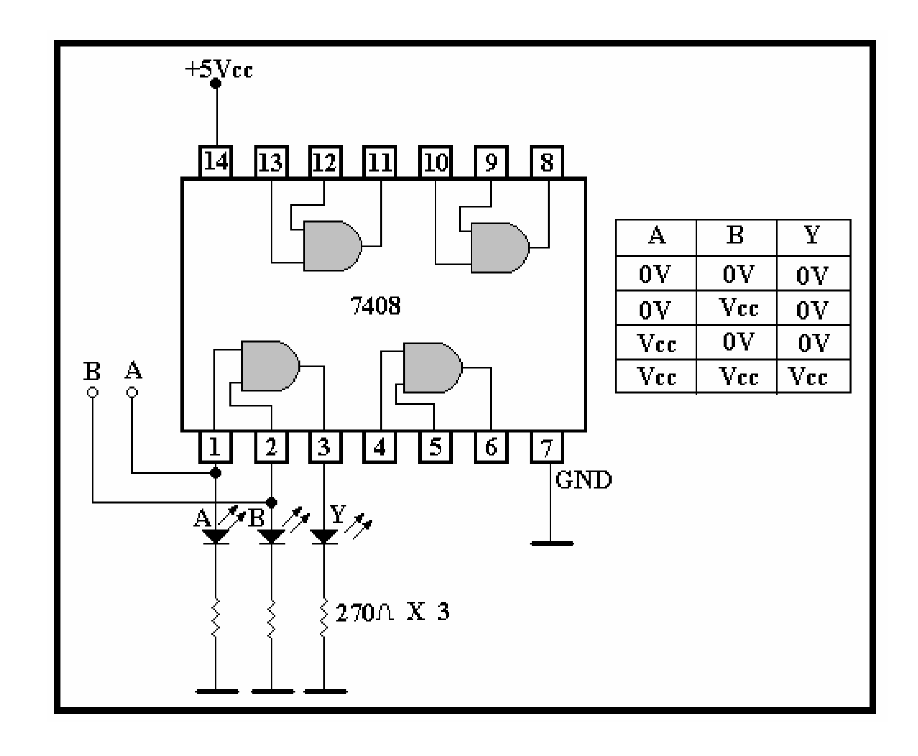
e) Conecte la entrada A de la primera compuerta a “1 lógico” y la entrada B a “0 lógico”. Compruebe el nivel de salida y anótelo en la tabla d e la verdad.

f) Conecte la entrada A de la primera compuerta a “0 lógico” y la entrada B a “1 lógico”. Averigue el nivel de la salida y anótelo en la tabla de la verdad.

g) Conecte ambas entradas a “1 lógico” y, posteriormente, verifique el nivel de salida y anótelo en la tabla de la verdad.

**Conclusiones:**..........................................................................................................................
........................................................................... ..............................................................
..........................................................................................................................................

**4.-Verificar tabla de la verdad del CI 7408, que actúa como cuadruple compuerta AND de dos entradas cada una.**




| A | B | Y |
|---|---|---|
| 0V | 0V | 0V |
| 0V | Vcc | 0V |
| Vcc | 0V | 0V |
| Vcc | Vcc | Vcc |

a) Inserte el CI 7408 entre dos columnas anchas del protoboard.

b) Regule la fuente a 5Vcc.

c) Alimente el pin 14 del integrado con +5Vcc y el pin 7 con el negativo de l a fuente

d) Conecte la entrada A de la primera compuerta "0 lógico" y la entrada B a "0 lógico". Luego, a través de la punta de prueba lógica verifique el nivel de salida y anótelo en la tabla de la verdad.

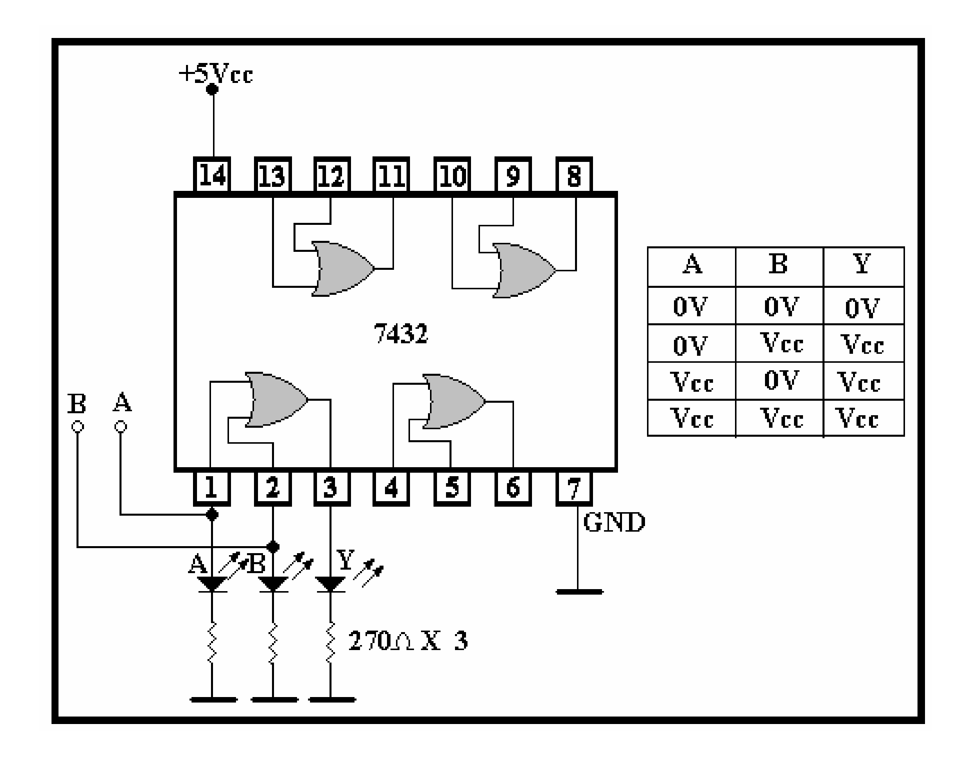
e) Conecte la entrada A de la primera compuerta a "0 lógico" y la entrada B a "1 lógico". Luego, verifique el nivel de salida y anótelo en la tabla de la verdad.

f) Conecte ambas entradas (A y B) a "1 lógico" y luego, a través de la punta de prueba lógica verifique el nivel de salida y anótelo en la tabla de l a verdad.

g) Repita estas pruebas en cada una de las restantes compuertas que conforman al integrado 7408.

**Conclusiones:**.........................................................................................................
.........................................................................................................................

**5.-<u>Verificar tabla de la verdad del CI 7432, que actúa como cuadruple compuerta OR de dos entradas cada una.</u>**




| A | B | Y |
|---|---|---|
| 0V | 0V | 0V |
| 0V | Vcc | Vcc |
| Vcc | 0V | Vcc |
| Vcc | Vcc | Vcc |

a) Inserte el CI 7432 entre dos columnas anchas del protoboard.

b) Regule la fuente a 5Vcc.

c) Alimente el pin 14 del integrado con +5Vcc y el pin 7 con el negativo de la fuente.

d) Conecte la entrada A de la primera fuente a “0 lógico” y lo mismo haga co n la entrada B verifique posteriormente el nivel de entrada y anótelo en la tabla de la verdad.

e) Conecte la entrada A de la primera compuerta a “1 lógico” y la entrada B a “0 lógico”. Compruebe el nivel de salida y anótelo en la tabla de la verdad.

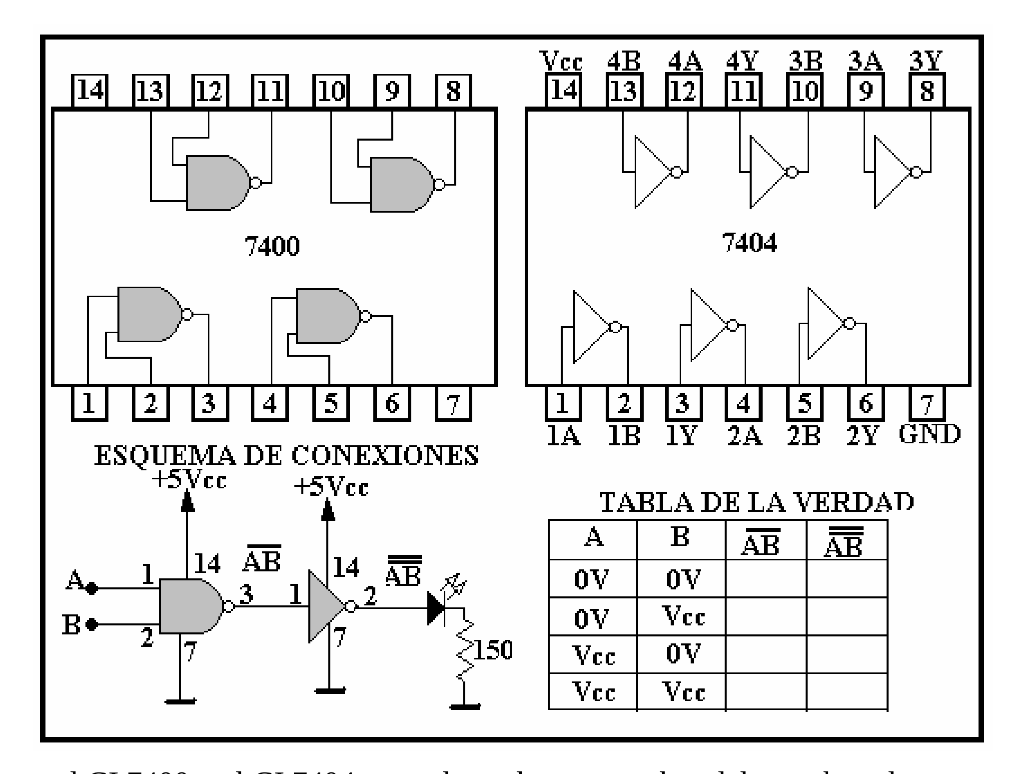
f) Conecte la entrada A de la primera compuerta a “0 lógico” y la entrada B a “1 lógico”. Averigue el nivel de la salida y anótelo en la tabla de la verdad.

g) Conecte ambas entradas a “1 lógico” y, posteriormente, verifique el nivel de salida y anótelo en la tabla de la verdad.

**Conclusiones:**.......................................................................................................................
............................................................................................................ ............................................
..........................................................................................................................................

## PRACTICA DE LOGICA COMBINACIONAL

**1)Utilice un integrado 7400 y un integrado 7404, para conformar una compuerta AND de dos entradas.**




| A | B | $\overline{AB}$ | $\overline{\overline{AB}}$ |
|---|---|---|---|
| 0V | 0V | | |
| 0V | Vcc | | |
| Vcc | 0V | | |
| Vcc | Vcc | | |

a)Inserte el CI 7400 y el CI 7404 entre dos columnas anchas del protoboard.

b)Conecte la salida de la primera compuerta NAND (pin 3) a la entrada de la primer a compuerta inversora (pin 1). Ver esquema de conexiones.

c)Regule la fuente a 5Vcc

d)Alimente el pin 14 de ambos integrados con +5Vcc y el pin 7 de ambos integrados al negativo de la fuente

e)Conecte la entrada A de la primera compuerta NAND (pin 1 d el CI 7400) a "0 lógico" y la entrada B del mismo integrado (pin 2), también a "0 lógico". Compruebe el nivel de salida de la compuerta NAND (pin 3 del CI 7400) y anotelo en la tabla de la verdad.
Compruebe luego el nivel de salida a la salida de la compu erta inversora (pin 2 del CI 7404) y anótelo en la tabla de la verdad.

f)Conecte la entrada A de la primera compuerta NAND (pin 1 del CI 7400) a "1 lógico y la entrada B del mismo integrado (pin 2) a "0 lógico".

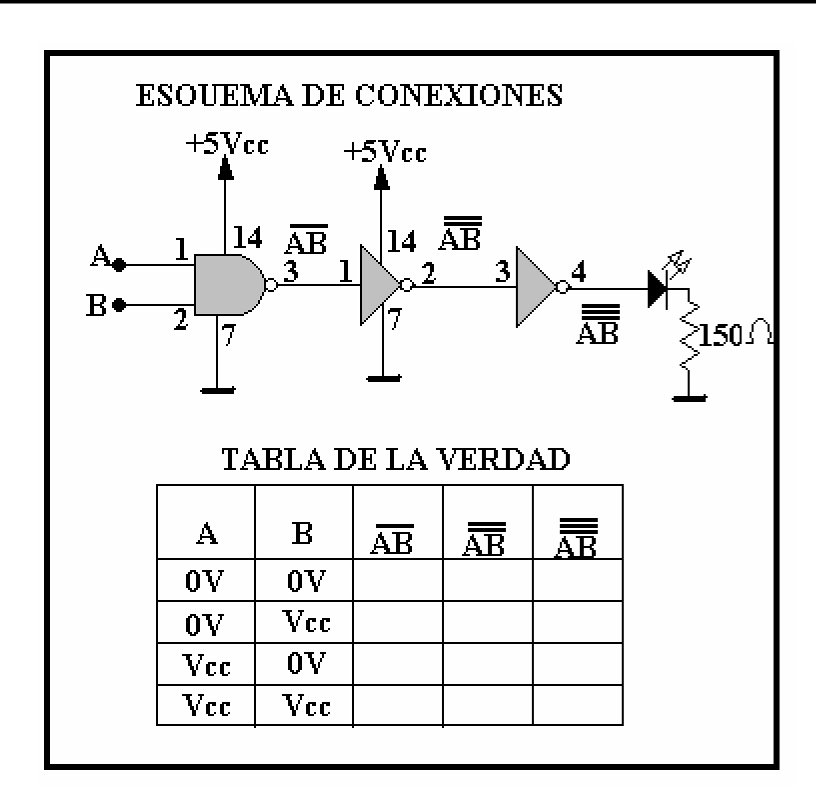
Compruebe luego el nivel de salida a la co mpuerta NAND y señálelo en la tabla de la verdad. Verifique posteriormente el nivel de salida de la compuerta inversora (pin 2 del CI 7404) y marquelo en la tabla de la verdad.

g)Conecte la entrada A de la primera compuerta NAND a "0 lógico" y la entrada B del mismo integrado a "1 lógico". Verifique luego el nivel de salida de la primera compuerta NAND (pin 3) y destaquelo en la tabla de la verdad. Posteriormente señale el nivel de salida de la primera compuerta inversora (pin 2) y destaquelo en la tabl a de la verdad.

h)Conecte ambas entradas de la primera compuerta NAND a "1 lógico" y verifique el nivel de salida de la misma (pin 3). Verifique luego, el nivel de salida de la primera compuerta inversora y destaquelo en la tabla de la verdad.

**Conclusión:**.......................................................................................................................
.......................................................................................................................... ...................
.........................................................................................................................................

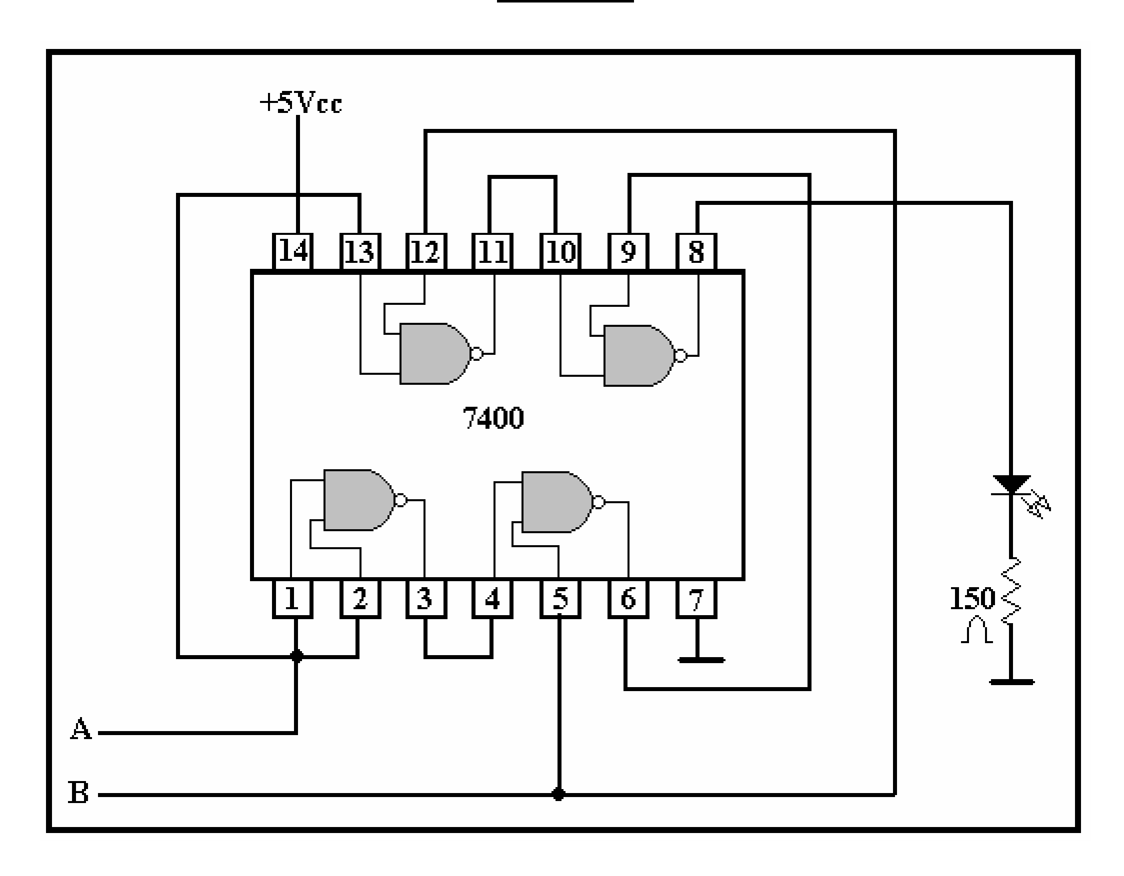
**2)Realice una tabla de la verdad en la cual se señale el nivel de salida resultante, si al mismo circuito se le agrega una nueva compuerta inversora a la salida.**




| A | B | $\overline{AB}$ | $\overline{\overline{AB}}$ | $\overline{\overline{\overline{AB}}}$ |
|---|---|---|---|---|
| 0V | 0V | | | |
| 0V | Vcc | | | |
| Vcc | 0V | | | |
| Vcc | Vcc | | | |

**Conclusión:** El circuito señalado es equivalente a una compuerta :..................

**3)Realice el siguiente montaje utilizando un CI 7400, un diodo led y u na resistencia de 150 ohms.**




a)Inserte el CI 7400 entre dos columnas anchas del protoboard.

b)Efectué el conexionado según lo indica el circuito.

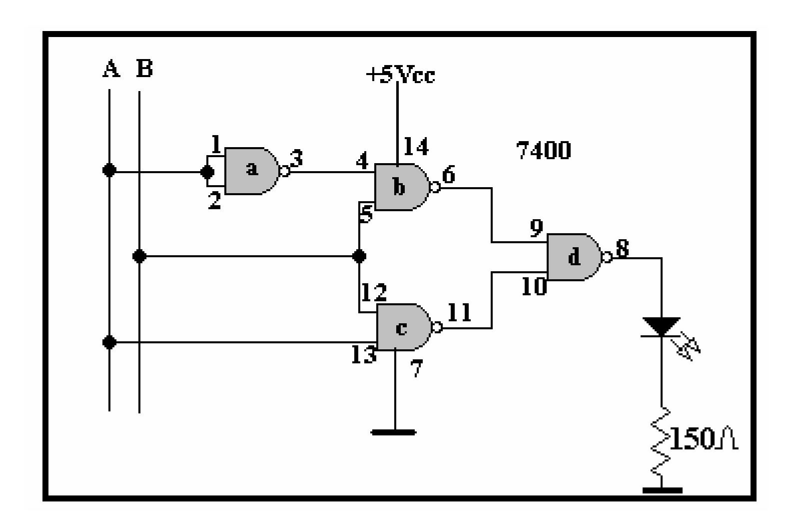
c)Conecte el diodo led y el resistor de 150 ohms.

d)Asegurese de que el conexionado se encuentra correcto.

e)Desarrollo la tabla de la verdad del circuito.

| A | B | $\overline{A}$ | AB | $\overline{A}\,B$ | $\overline{\overline{A}\,B}$ | $\overline{A\,B}$ | $\overline{(\overline{\overline{A}\,B})\,(\overline{A\,B})}$ |
|---|---|---|---|---|---|---|---|
| 0V | 0V | | | | | | |
| 0V | Vcc | | | | | | |
| Vcc | 0V | | | | | | |
| Vcc | Vcc | | | | | | |

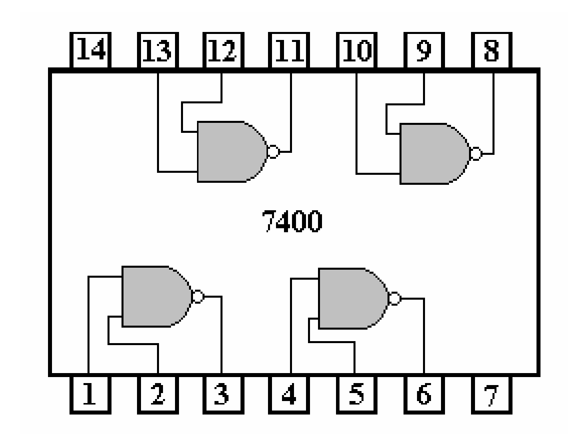
f)Esquematizar el circuito a través de compuertas lógicas. -

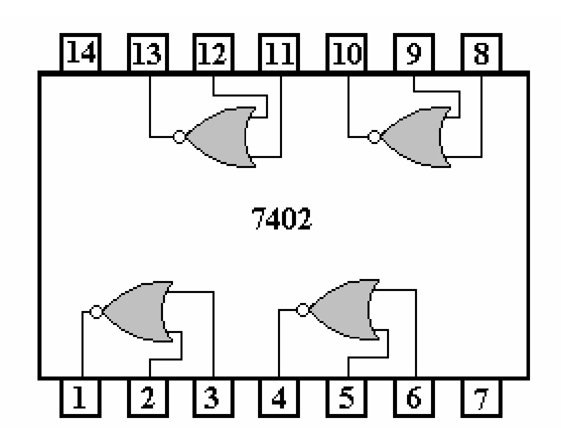



g)Comprobación del mismo a través de la punta lógica. -

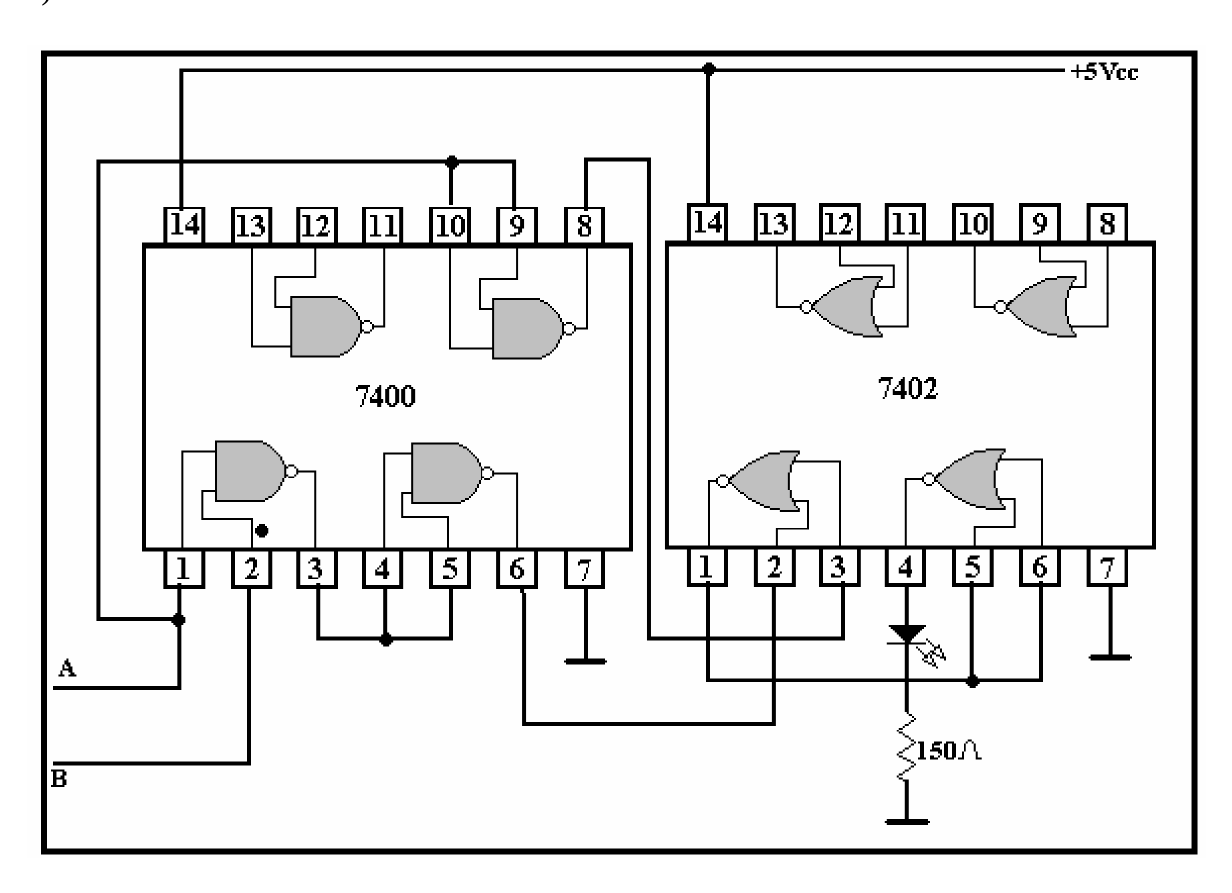
h)Verifique cada una de las salidas indicadas en la tab la de la verdad esquematizada más arriba y compárelas con dicha tabla.

**4)Utilice un CI 7400 y 7402, para realizar el siguiente montaje:**

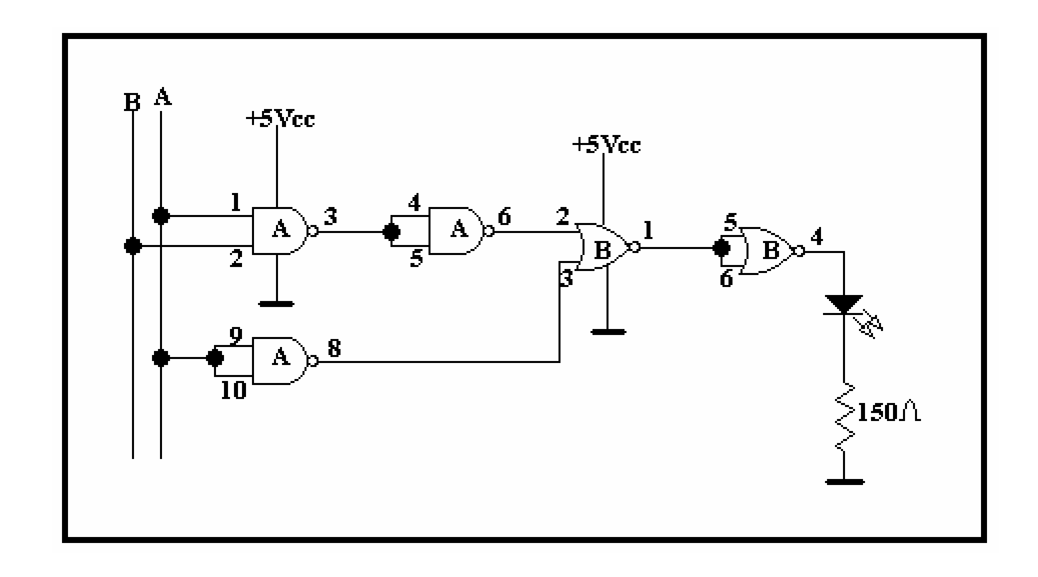


a)Circuito a realizar:




b)Circuito a desarrollar.-



d)Tabla de la verdad.-

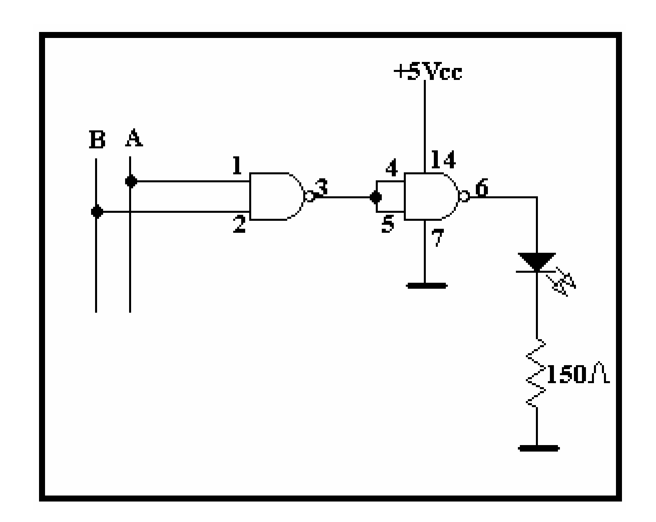
| B | A | Ā | A . B | A . B | A . B | A . B + Ā | A . B + Ā | A . B + Ā |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | | | | | | | |
| 0 | 1 | | | | | | | |
| 1 | 0 | | | | | | | |
| 1 | 1 | | | | | | | |

e)Montaje del circuito:

- a)Inserte los circuitos integrados entre dos columnas anchas del protoboard, a corta distancia entre sí.
- b)Efectúe el conexionado que señala en circuito superior.
- c)Agregue el diodo led y el resistor de 150 ohms.
- d)Asegurese de que el conexionado es el correcto.
- e)Verifique cada una de las salidas indicadas en la tabla de la verdad mo strada más arriba y compárelas con dicha tabla.

**5)Utilizando un CI7400, de la familia TTL, realice los siguientes conexionados:**

a)



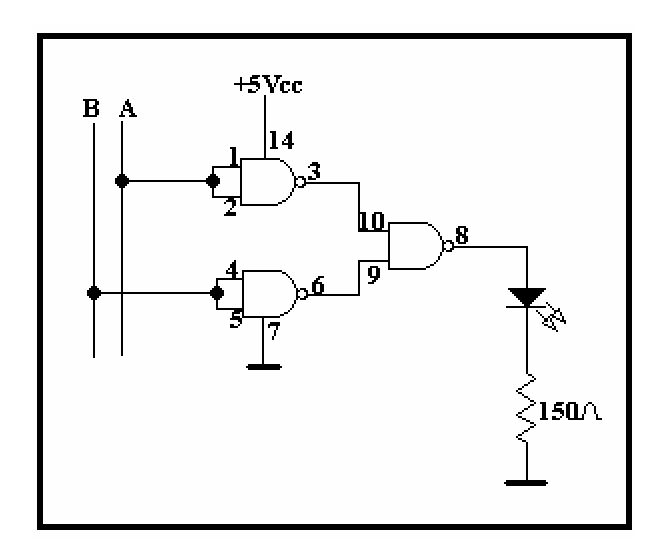
-Desarrolle una tabla de verdad del circuito:

-De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta .....................................de .............................................entradas.

-Dibuje la compuerta equivalente:

-A través de la punta de prueba lógica verifique si los niveles de salida del circuito coinciden con la tabla de la verdad.

b)



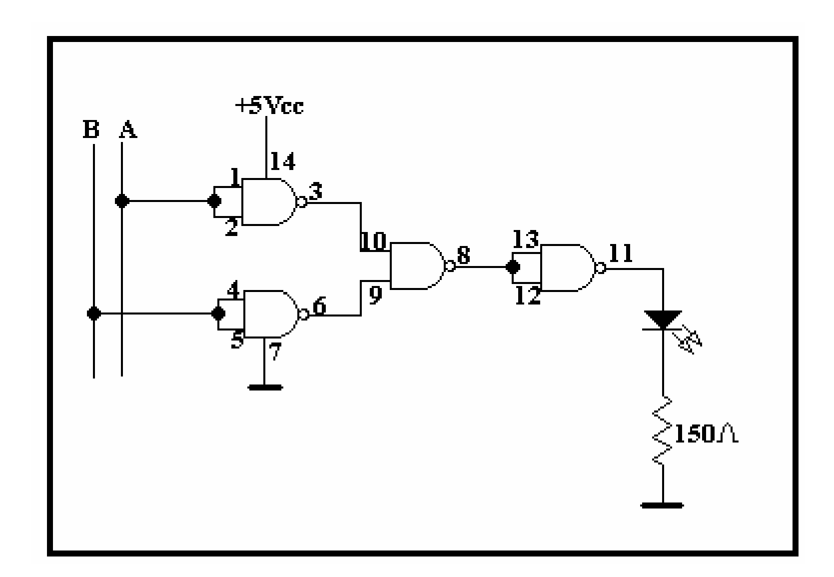

-Desarrolle la tabla de la verdad del circuito:

-De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta.....................................................de .............. .............................entradas.

-Dibuje la compuerta equivalente:

-Utilizando la punta de prueba lógica compruebe si los niveles de salida del circuito concuerdan con la tabla de la verdad.

c)




-Desarrolle la tabla de la verdad del circuito:

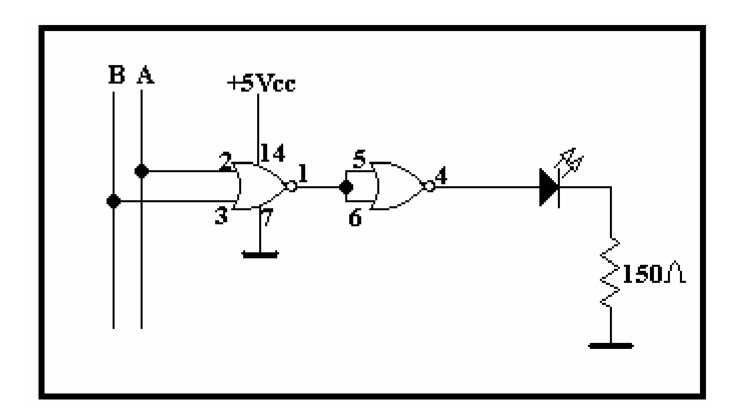
-En conformidad a la tabla de la verdad el circuito representado es equivalente a una compuerta....................................................de..................................................entradas.

-Dibuje la compuerta equivalente:

-A través de la punta de prueba lógica verifique si los niveles de salida del circuito concuerdan con la tabla de la verdad.

**<u>6)Utilizando un CI7402, de la familia TTL, realice los siguientes conexinados:</u>**

a)



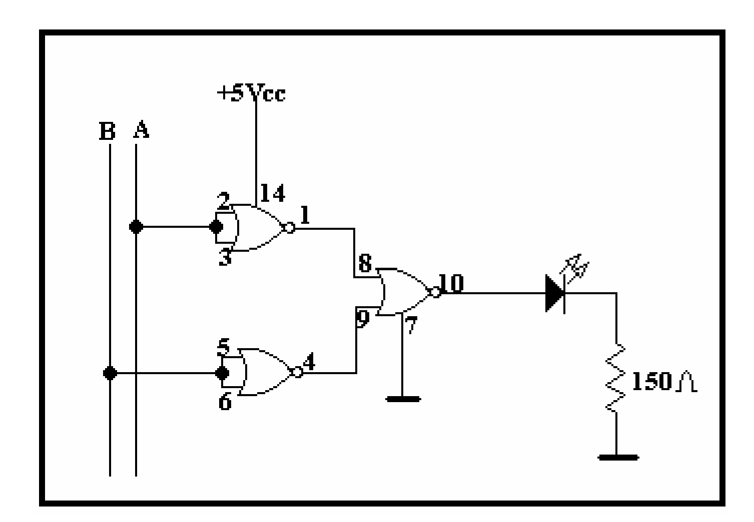
-Desarrolle la tabla de verdad del circuito:

-De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta.......................................................de..............................................entradas.

-Dibuje la compuerta equivalente:

-A través de la punta de prueba lógica certifique si los niveles de salida del circuito coinciden con la tabla de la verdad.

b)



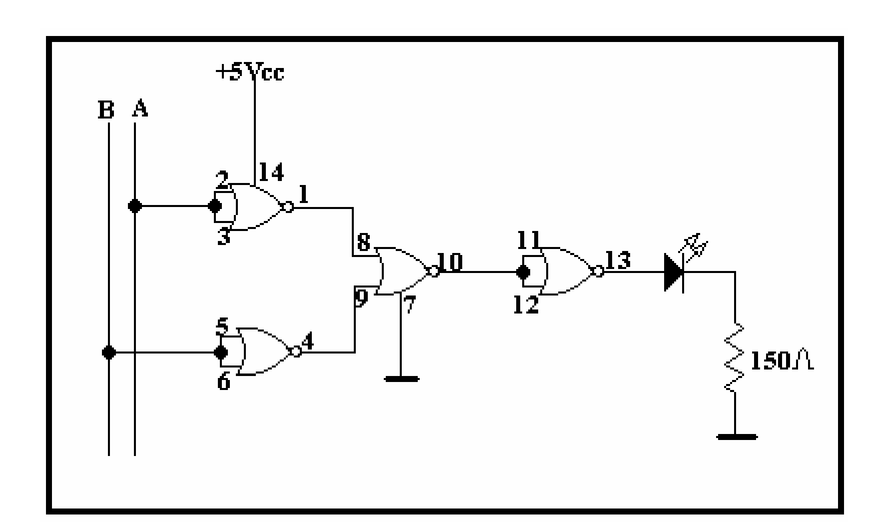

-Desarrolle la Tabla de la verdad del circuito:

-De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta...................................................de................................................entradas.

-Dibuje la compuerta equivalente:

-Por medio de la prueba lógica determine si los niveles de salida del circuito son coincidentes con la tabla de la verdad.

c)




-Desarrolle la tabla de la verdad del circuito:

-De acuerdo a la tabla de la verdad el circuito final es equivalente a una compuerta....................................... ........de......................................................entradas.

-Dibuje la compuerta equivalente:

-A través de la punta de prueba lógica determine si los niveles de salida del circuito coinciden con la tabla de la verdad.